



LiveCycle® ES Overview

July 2007

Adobe® LiveCycle® ES

Version 8.0

© 2007 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle® ES 8.0 Overview for Microsoft® Windows®, Linux®, and UNIX®
Edition 1.1, July 2007

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names, company logos and user names in sample material or sample forms included in this documentation and/or software are for demonstration purposes only and are not intended to refer to any actual organization or persons.

Adobe, the Adobe logo, Acrobat, ColdFusion, Distiller, Flash, Flex, Flex Builder, LiveCycle, PostScript, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

BEA WebLogic Server is a registered trademark and BEA WebLogic Platform is a trademark of BEA Systems, Inc.

EMC and Documentum are registered trademarks of EMC Corporation in the United States and around the world. Copyright 1994-2007 EMC Corporation, all rights reserved.

IBM, FileNet, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java, JavaBeans, JavaScript, and JavaServer Pages are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries

JBoss is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

All other trademarks are the property of their respective owners.

This product contains either BISAFE and/or TIPEM software by RSA Data Security, Inc.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes code licensed from RSA Data Security.

This product includes software developed by the JDOM Project (<http://www.jdom.org/>).

Macromedia Flash 8 video is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved.
<http://www.on2.com>.

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>).

Portions of this code are licensed from Nellymoser(www.nellymoser.com).

MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and THOMSON Multimedia (<http://www.iis.fhg.de/amm/>).

This product includes software developed by L2FProd.com (<http://www.L2FProd.com/>).

The JBoss library is licensed under the GNU Library General Public License, a copy of which is included with this software.

The BeanShell library is licensed under the GNU Library General Public License, a copy of which is included with this software.

This product includes software developed by The Werken Company (<http://jaxen.werken.com/>).

This product includes software developed by the IronSmith Project (<http://www.ironsmith.org/>).

The OpenOffice.org library is licensed under the GNU Library General Public License, a copy of which is included with this software.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

About This Document.....	6
Who should read this document?	6
Additional information.....	6
1 About Adobe LiveCycle ES	7
Key components.....	7
LiveCycle Foundation.....	8
Solution components.....	8
Development tools.....	9
Architecture.....	10
Typical deployment scenarios.....	11
Single node deployment.....	11
Clustered volume production deployment	12
2 LiveCycle Foundation.....	14
Process coordination and service management	14
Job management and monitoring	14
Repository and service registry	15
Registering assets and services	15
Creating LiveCycle ES applications	15
Service registry	16
Common invocation support.....	16
User management and authentication	17
LiveCycle Administration Console	17
3 LiveCycle ES Development Tools	18
Developing processes and forms	18
Designing processes	18
Designing forms	18
Developing applications	19
Developing rich Internet applications	19
LiveCycle ES SDK.....	19
4 LiveCycle Process Management ES	20
Key features	20
Task assignment and management	20
Event management.....	20
LiveCycle Workspace ES	21
Services included with Process Management ES.....	22
5 LiveCycle Business Activity Monitoring ES.....	23
Key features	23
Analytics server.....	23
Performance dashboards.....	24
Analytical workbench.....	24

6 LiveCycle Data Services ES.....	25
Key features	25
How Data Services ES works.....	26
Using Data Services ES	27
LiveCycle Remoting.....	27
Message Service	27
Data Management Service	27
RPC services	27
Flex-Ajax Bridge.....	28
Ajax Data Services.....	28
RIA-to-PDF generation	28
7 LiveCycle Forms ES	29
Key features	29
Rendering forms and integrating form data.....	29
Creating forms using form guides.....	29
Rendering forms based on fragments.....	30
Assembling PDF documents and forms	30
How Forms ES works.....	31
Services included with Forms ES	32
8 LiveCycle Output ES.....	33
Key features	33
Document design for Output ES	33
Supported document formats	34
Controlling print features	34
Generating flexible documents	34
Document conversion and assembly	34
Document and process automation	35
How Output ES works.....	35
Services included with Output ES	36
9 LiveCycle Reader Extensions ES	37
Key features	37
How Reader Extensions ES works.....	38
Services included with Reader Extensions ES	39
10 LiveCycle Barcoded Forms ES	40
Key features	40
How Barcoded Forms ES works.....	41
Authoring barcoded forms.....	42
Creating a process	42
Adding barcoded forms usage rights for Adobe Reader	42
Updating barcodes during form completion	43
Decoding barcodes to extract barcode data	43
Processing captured barcode data	43
Services included with Barcoded Forms ES	43
11 LiveCycle Digital Signatures ES	44
Key features	44
How Digital Signatures ES secures a document	44
Services included with Digital Signatures ES	45

12 LiveCycle Rights Management ES	46
Key features	46
How Rights Management ES secures a document	47
About Rights Management ES security	48
Services included with Rights Management ES	49
13 LiveCycle PDF Generator ES.....	50
Key features	50
How PDF Generator ES works	51
Document types suitable for conversion	51
Postscript and EPS to PDF	51
Native file formats to PDF	52
PDF to other format	52
Input file formats	52
Open standards support.....	53
Conversion from PDF to other formats	53
Services included with PDF Generator ES	53
14 LiveCycle ES Connectors for ECM	54
Key features	54
Integrated content repository services.....	54
Extending Workbench ES.....	55
Improved performance and scalability	55
Flexible deployment	55
Unified invocation methods	55
How the LiveCycle ES connectors for ECM work	56
Services included with connectors for ECM	57

About This Document

Adobe® LiveCycle® ES (Enterprise Suite) enables organizations to automate and accelerate the flow of business-critical information among employees, customers, suppliers, and constituents.

This document provides an overview of LiveCycle ES (Enterprise Suite), including information about the services that the licensable solution components provide.

Who should read this document?

This document provides information for business analysts, form authors, process developers, and programmers who need to find out how they can use LiveCycle ES for designing business processes, creating forms, or developing process diagrams or client applications to invoke the business processes. This document is also useful to administrators who plan to install, configure, or administer LiveCycle ES.

Additional information

The resources in this table can help you learn more about LiveCycle ES.

For information about	See
What's new in Adobe LiveCycle ES	http://www.adobe.com/go/learn_lc_whatsNew
Preparing to install and deploy LiveCycle ES	http://www.adobe.com/go/learn_lc_prepareInstall
Installing and deploying LiveCycle ES	http://www.adobe.com/go/learn_lc_installTurnkey or http://www.adobe.com/go/learn_lc_installJBoss http://www.adobe.com/go/learn_lc_installWebSphere http://www.adobe.com/go/learn_lc_installWebLogic
Installing and configuring Workbench ES	http://www.adobe.com/go/learn_lc_installWorkbench
How to create your first LiveCycle ES application	http://www.adobe.com/go/learn_lc_firstApplication
Service descriptions	http://www.adobe.com/go/learn_lc_services
LiveCycle ES terminology	http://www.adobe.com/go/learn_lc_glossary
Other services and products that integrate with LiveCycle ES	http://www.adobe.com
Patch updates, technical notes, and additional information on this product version	http://www.adobe.com/support/products/enterprise/index.html

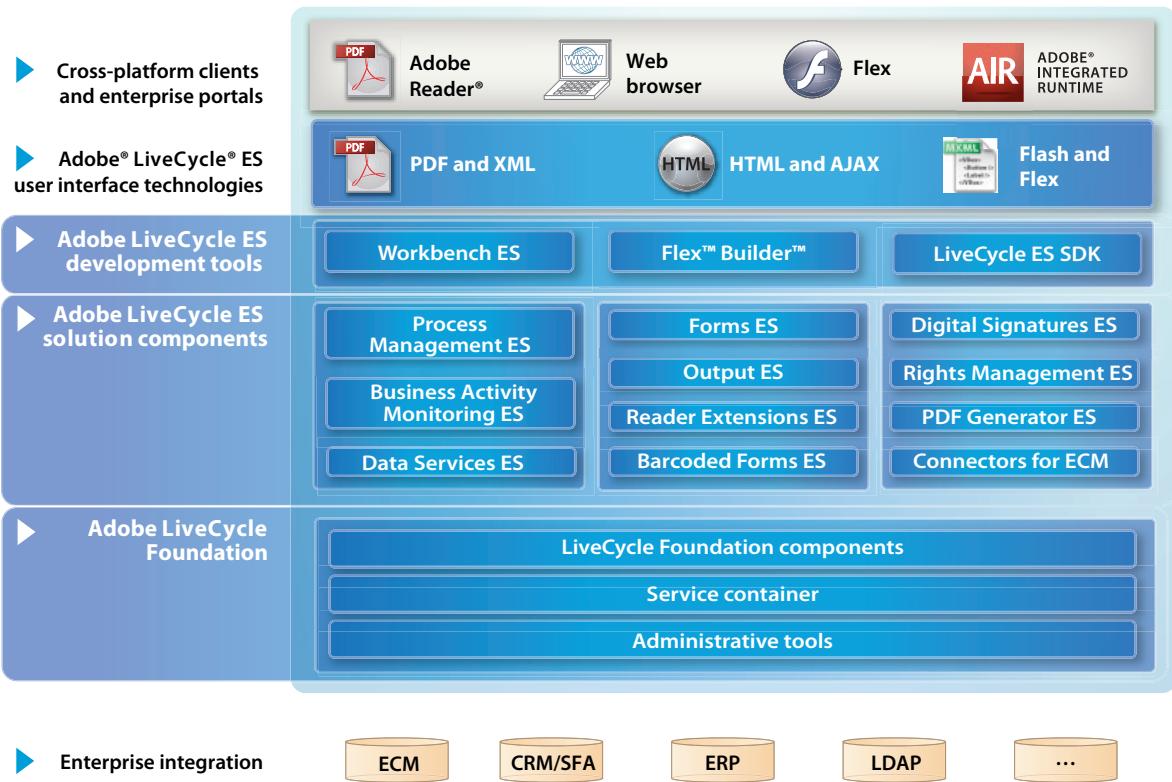
Adobe LiveCycle ES (Enterprise Suite) is an enterprise server platform that enables your organization to automate and streamline business processes that span systems, business rules, and departments. It provides all the tools you need to design, implement, and automate the documents and processes that you use within your organization and with external customers and stakeholders. LiveCycle ES combines Adobe PDF and Adobe Flex™ technology to provide a unified developer experience for creating customer engagement applications—applications that revolutionize the user experience.

Adobe LiveCycle Workbench ES is an Eclipse™-based development environment that lets you create and manage business processes and the forms used in those processes. In a production environment, processes invoke the services that render forms, assemble PDF documents, apply security to protect documents that are sent beyond the firewall, output documents in a variety of formats, and apply usage rights to extend their functionality in Adobe Reader®. Services enable web-based customer interaction by allowing end users to participate in business processes through an intuitive web interface or to capture data that customers submit through rich Internet applications (RIAs).

Key components

LiveCycle ES provides the resources that business analysts, form and application developers, and administrators need in order to create and deliver applications that support user interaction in cross-platform technologies, including Adobe Reader, HTML, and Adobe Flash®.

LiveCycle ES consists of three major components: LiveCycle Foundation, solution components, and development tools.



LiveCycle Foundation

Adobe LiveCycle Foundation provides the underlying server capabilities that enable the deployment, execution, and management of solution components. LiveCycle Foundation consists of several pieces.

LiveCycle Foundation components

Components that enable the LiveCycle ES server to integrate with a common IT infrastructure. For example, these components enable tasks that include querying a directory through LDAP, sending and receiving email, sending and receiving messages over a Java™ Message Service (JMS) queue, querying a relational database, and writing files to the file system.

Service container

The service container provides the common run-time environment to support all solution components and associated services.

Administration tools

LiveCycle Foundation includes several administration tools:

LiveCycle Administration Console: A web-based interface that system administrators use to manage a LiveCycle ES deployment. The key features are as follows:

- Administration pages for solution components, including web interfaces for Adobe LiveCycle Process Management ES, Adobe LiveCycle Rights Management ES, Adobe LiveCycle PDF Generator ES, Adobe LiveCycle Forms ES, and Adobe LiveCycle Output ES
- Configuring server settings, such as port numbers
- Configuring user groups, roles, and associated permissions
- Deploying and configuring LiveCycle ES applications.

See ["LiveCycle Administration Console" on page 17](#).

LiveCycle Configuration Manager: Enables the configuration and deployment of the product, including adding service packs and patches.

Solution components

Solution components provide the functional services needed to build customer engagement applications that can be deployed to the LiveCycle ES server. For example, solution components offer business functions such as capturing data in form guides (which is based on the Flash technology), encrypting PDF documents, applying policies or Adobe Reader usage rights to PDF documents, converting an image to PDF, and remote invocation of services.

Development tools

LiveCycle ES provides development tools that allow a wide variety of users, from business analysts to Java 2 Platform, Enterprise Edition (J2EE) developers, to collaborate on the design, development, testing, and deployment of a LiveCycle ES application.

Workbench ES: An Eclipse-based development environment that lets users build applications that consist of forms and business processes. Form authors and developers can create forms that have either a *fixed layout* (the layout remains exactly as it was designed; it does not change to accommodate the amount of incoming data) or a *flowable layout* (the layout expands or shrinks to accommodate the amount of data being merged or entered). These forms can be rendered using a number of client technologies such as PDF, HTML, and Flash. Business analysts and process developers can use Workbench ES to design and deploy business processes that leverage the solution components.

Flex Builder: Adobe Flex Builder™ 2 is an Eclipse-based development environment for developing rich Internet applications (RIAs) with the Flex framework. Using Flex Builder 2, developers can quickly build and deploy applications that are expressive, intuitive, and interactive. Flex Builder lets developers customize their applications to ensure a look and feel that represents the needs of end users.

LiveCycle ES SDK: The LiveCycle ES SDK provides developers with a set of samples and tools that they can use to build new LiveCycle ES applications or integrate with existing applications.

For more information, see these topics:

- ["LiveCycle Process Management ES" on page 20](#)
- ["LiveCycle Business Activity Monitoring ES" on page 23](#)
- ["LiveCycle Data Services ES" on page 25](#)
- ["LiveCycle Forms ES" on page 29](#)
- ["LiveCycle Output ES" on page 33](#)
- ["LiveCycle Reader Extensions ES" on page 37](#)
- ["LiveCycle Barcoded Forms ES" on page 40](#)
- ["LiveCycle Digital Signatures ES" on page 44](#)
- ["LiveCycle Rights Management ES" on page 46](#)
- ["LiveCycle PDF Generator ES" on page 50](#)
- ["LiveCycle ES Connectors for ECM" on page 54](#)

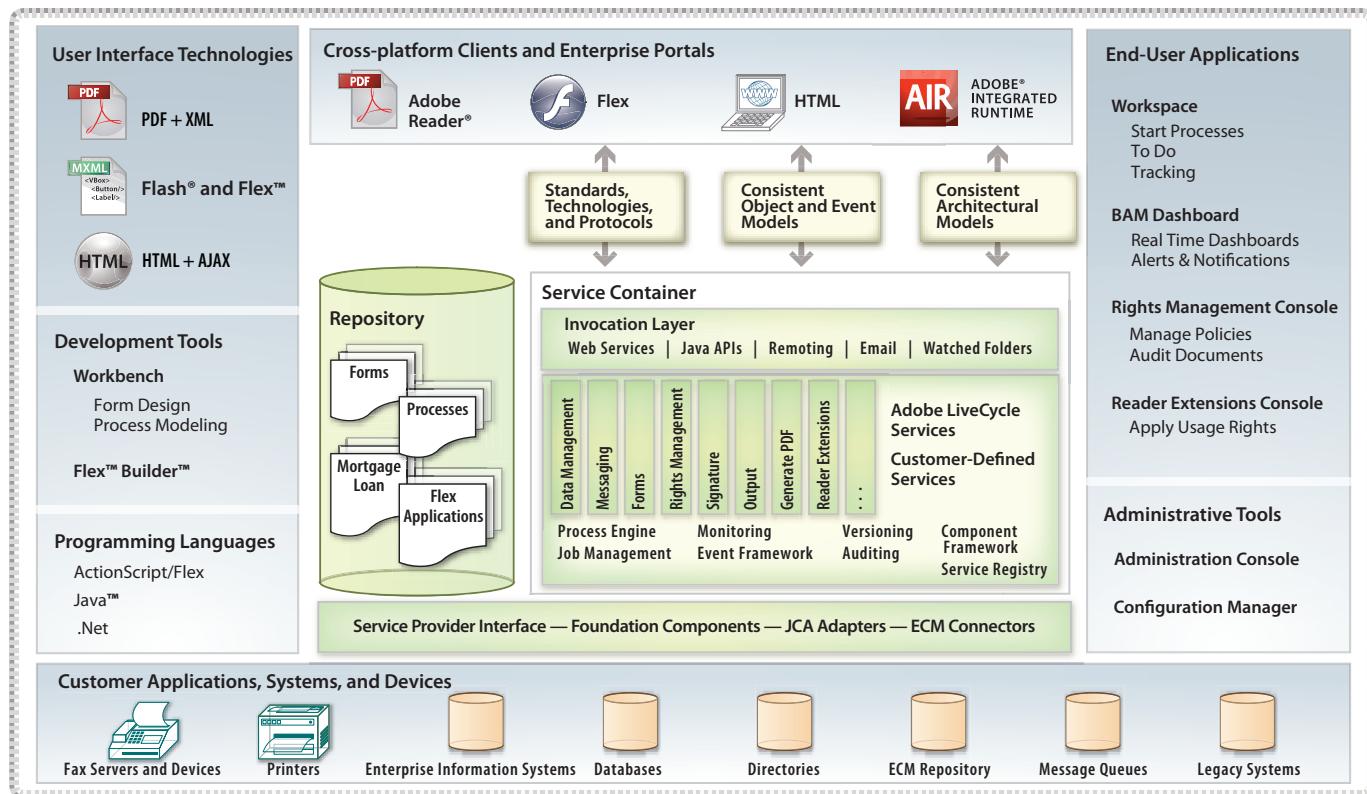
LiveCycle Foundation and solution components are installed and deployed using a single installation and deployment framework. The resulting suite integrates into a unified solution, which plugs into an enterprise back-end environment and supports a variety of cross-platform clients for end-user interaction.

Architecture

LiveCycle ES implements a service-oriented architecture that relies on loosely coupled software services to support business processes and perform operations on documents. Services run within the service container, which hosts any number of services and provides the infrastructure necessary to discover, execute and deploy, secure, invoke, and configure the services.

The service container provides a consistent way to interact with the services within the container, the repository, and the enterprise integration, regardless of the invocation method. Services can be invoked using Java APIs, web services, watched folders, or email. Service endpoints are also exposed as destinations that can easily be invoked by RIAs.

This illustration provides a closer view of how the key product components fit together within the server architecture.



The service container is an extensible components model based on SOA principles: components can be added, upgraded, reused, or removed with minimal administration. Components are independent units of functionality that plug into the service container for the purpose of deploying, orchestrating, and executing services. The service container coordinates and monitors the components.

The development area includes Workbench ES tools and the repository. Development of forms and form guides, process flows, and related collateral is done in Workbench ES. The Eclipse-based development environment can also include Flex Builder (available separately), which you can use to develop RIAs, create custom components for use in form guides, customize Adobe LiveCycle Workspace ES, and more. For information about Workspace ES, see ["LiveCycle Workspace ES" on page 21](#).

The resulting design artifacts are stored in the repository, which provides versioning and maintains resource dependencies. This central storage and maintenance model promotes the reusability of artifacts, enables developers to collaborate on application development, and provides security within the development environment.

The common invocation layer ensures consistent interaction with the server through a variety of invocation methods. It supports programmatic and non-programmatic invocation methods, including web services, Java APIs, watched folders, LiveCycle Remoting, and email-based invocation methods. All of these methods of invocation are available to use with any service.

LiveCycle ES provides integration adapters to ensure compatibility with your enterprise back-end systems. These back-end systems include user directories, authentication systems, Enterprise Content Management (ECM) systems, web services, Java Message Service (JMS), Java Remote Method Invocation (RMI), and email.

The service-oriented architecture of LiveCycle ES maximizes the scalability and reusability of services. It ensures that you can easily add, remove, and upgrade services. User authentication, service deployment, and invocation are all managed within a common server infrastructure to ensure system security and performance.

Typical deployment scenarios

The deployment possibilities for LiveCycle ES are flexible; you can deploy it as a single stand-alone server running a single or multiple solution components, or as a volume production system running multiple solution components on clustered servers.

This section describes the following deployment scenarios:

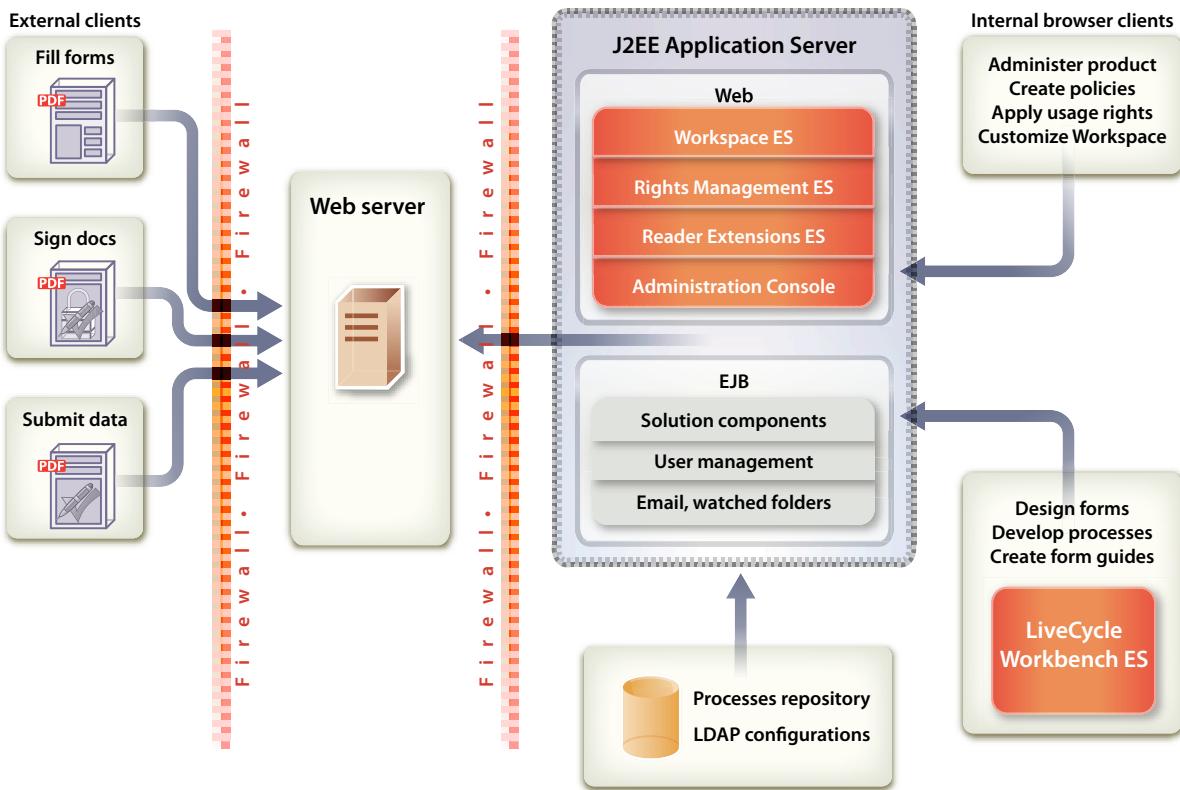
- Single node deployment for a small production system
- Clustered deployment for volume production in an enterprise environment

Single node deployment

A single node deployment is useful for running simple jobs. For example, Adobe LiveCycle PDF Generator ES can be deployed to a single node for converting Microsoft® Office documents to PDF documents.

You can install and deploy LiveCycle ES on a single application server instance by using the turnkey method. This method installs and deploys an instance of the JBoss® Application Server, MySQL database server, and deploys LiveCycle ES and licensed components to the application server as a complete and ready-to-use application.

You can configure and deploy LiveCycle ES automatically to IBM® WebSphere® Application Server or BEA WebLogic Server® if they are already set up. In this case, you also need to install a database server. You can install the database, as well as Workbench ES or the LiveCycle ES SDK, on the same or a separate computer.



The structure of the single node deployment is replicated in a clustered enterprise deployment but on a larger scale.

Clustered volume production deployment

The structure of the single node deployment is replicated in a clustered enterprise deployment but on a larger scale. In larger scale deployments, database and LDAP servers are typically located remotely, and a more rigorous distinction between development, testing, and production environments is implemented.

In a clustered large-volume production infrastructure, the system is set up so that several people play different roles in the system. For example, an IT administrator installs and deploys the production system (and possibly the testing system), and sets up the security infrastructure. Form developers design forms and form guides; process developers design processes; administrators deploy services, manage server performance, and so on. The system integrates with the organization's back-end database infrastructure.

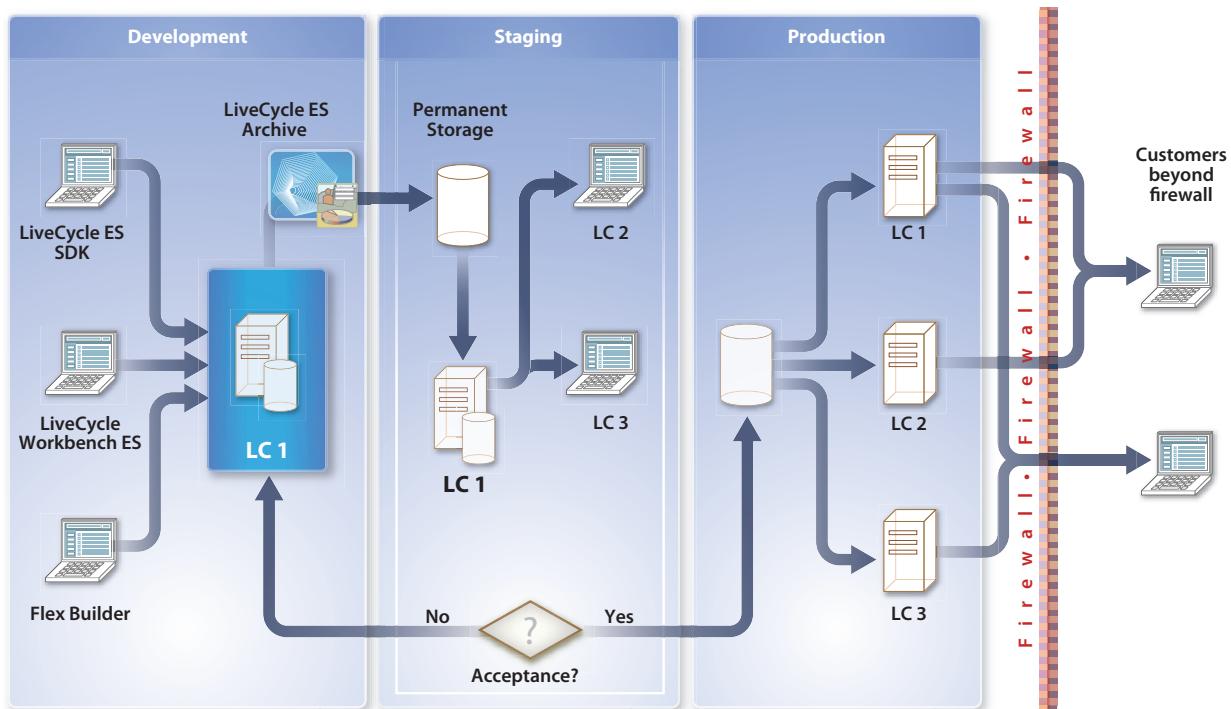
Typically, various users use LiveCycle ES in three phases of environments:

Development environment: Form authors, process developers, or programmers use the development and programming tools to create processes and custom applications for deployment.

Staging environment: Administrators and testers deploy and test the applications in a system that mimics the final production deployment scenario.

Production environment: Administrators deploy, monitor, and maintain services and applications. End users interact with services within and outside your organization (and within and outside the firewall).

At each of these phases, you must move all of the various assets, such as PDF forms, images, process diagrams, and other files necessary for an application to work from one phase to the next. Developers typically package all collateral associated with an application into a LiveCycle ES archive (LCA) file and transfer the application from development to staging to production.



Adobe LiveCycle Foundation provides the underlying server capabilities on which LiveCycle ES solutions can be built. Included in every LiveCycle ES installation, LiveCycle Foundation provides all the tools required for application development, invocation, management, and administration. LiveCycle Foundation integrates with your organization's existing infrastructure of databases and LDAP systems, and works to assemble the services that are invoked by deployed processes.

LiveCycle Foundation includes a number of services and components that integrate to let you deploy and manage business documents and processes:

- Process coordination and service management
- Common invocation support
- User management
- LiveCycle Administration Console

Process coordination and service management

LiveCycle Foundation provides a common run-time environment, or service container, for all services that run in LiveCycle ES. This common environment facilitates process coordination, enabling developers to link multiple solution components in a process.

Job management and monitoring

The Job Manager service provides the capability to asynchronously invoke a component and retrieve the persisted results. It also provides the capabilities to monitor the execution of each invocation.

Using the Job Manager service API, developers can do these tasks:

- Create a new asynchronous job using the specified invocation request.
- Complete an existing job using the specified invocation response.
- Terminate, suspend, or resume an existing job identified by the specified job ID.
- Get the job ID that represents the status of a long-lived process. The job status indicates whether a job was queued, running, completed, terminated, suspended, or resumed. The status can also indicate whether a request was issued to complete, terminate, suspend, or resume a job.

For more information about invoking long-lived processes, see *Invoking LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_programming.

Repository and service registry

The repository provides the capability to manage the assets that developers create as part of their LiveCycle ES applications.

Developers can access the repository using the LiveCycle Form Design perspective in Workbench ES or programmatically using the repository API. A developer must be granted access before accessing the repository. Each time a developer uses Workbench ES, a connection to the repository is made. The repository is exposed as a hierarchical directory structure. One or more developers can share the repository from Workbench ES.

Staging and production systems each have their own repository. For example, an organization's quality assurance team tests a service in their staging environment. When the tests are successful, the team deploys the service into their production environment. When a service is deployed into production, the service has no dependency on any design-time assets in the staging environment's repository. The organization can optionally use the service registry's access control mechanisms to restrict access to the service deployed in the production environment. This enables the organization to pilot a deployment with a restricted group of users.

Registering assets and services

The repository provides storage capabilities. When a developer creates an application, the developer can deploy the assets in the repository instead of deploying them on a file system. The assets may consist of XML forms, PDF forms (including Acrobat forms), fragments, images, processes, profiles, policies, DDX files, XML schemas, WSDL files, SWF files, and test data.

The repository tracks the version of each asset in a LiveCycle ES application. At run time, services can retrieve assets from the repository as part of completing an automated business process.

Creating LiveCycle ES applications

The repository maintains dependency relationships among all of the assets it manages. LiveCycle ES uses these dependency relationships to assemble all the necessary assets into a LiveCycle ES application.

The application manager supports the capability to package the assets that are part of a LiveCycle ES application into a LiveCycle ES archive file. The archive file facilitates the transfer of an application from development to staging to production.

When a LiveCycle ES application is deployed, all of the assets within it are also deployed. The process of deploying some of those assets results in services being registered in the service registry and which can be invoked by the Invocation framework.

For example, when a process is deployed, a service entry is created in the service registry that will allow the process to be invoked as a service. If the service is published, a WSDL file is created and added to the service registry, along with the necessary metadata that the LiveCycle ES SDK framework uses to invoke the service.

Service registry

The service registry is used at run time to resolve the name of a service to an actual endpoint in the service container. Many different versions of a service can be deployed at any one time in the service registry. The Invocation framework, along with version information provided by the calling application, is used to bind the correct version of the service.

Services require a service container to run, similar to how Enterprise JavaBeans™ (EJBs) require a J2EE container. LiveCycle ES includes only one implementation of a service container, which is responsible for managing the lifetime of a service, including deploying it and ensuring that all requests are sent to the correct service. The service container is also responsible for managing documents that are consumed and produced by a service.

For more information about the service container, see *Invoking LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_programming.

Common invocation support

LiveCycle Foundation provides a common invocation layer, which ensures consistent interaction with the server, regardless of the invocation methods used. The LiveCycle Foundation invocation layer supports programmatic and non-programmatic invocation methods, including web services, Java APIs, watched folders, LiveCycle Remoting, and email-based invocation methods. All of these methods are available to invoke any service.

Services can be programmatically invoked from client applications that are developed by using a Java integrated development environment (IDE) or from a Flex or Ajax RIA by using Adobe LiveCycle Data Services ES.

To develop a client application in a Java development environment, you use Java APIs. LiveCycle ES also enables client applications to invoke its services by using web services.

Invocation API: A Java API that can be used to programmatically invoke any service. Use the Invocation API to invoke services, such as coordinate services that do not have strongly-typed APIs.

Strongly-typed Java API: A Java API that is used to invoke a specific service. A strongly-typed API is known as a service client and is used only to invoke a specific service. That is, you cannot use a service client that belongs to one service to invoke another service. These APIs can use Remote Method Invocation (RMI) or SOAP as the communication protocol between the client and the LiveCycle ES server.

Web services: Services located in the service container can be configured to expose a web service, with full support for Web Services Definition Language (WSDL) generation. You can create a proxy library from any service's WSDL. Using the proxy library, you can invoke a service.

Watched folders: A service can be invoked from a network folder that an administrator has configured as a watched folder through the LiveCycle Administration Console. When a file is placed in the folder, a service operation that manipulates the file is invoked.

Email: A service can be invoked when a configured email account receives an email message, typically with a PDF document as an attachment. A LiveCycle ES administrator configures the email account details through the LiveCycle Administration Console. After LiveCycle ES performs the operation, it sends an email message to the recipient with a modified PDF document attached.

For more information about invoking services, see *Invoking LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_programming.

For more information about configuring watched folders and email accounts for invoking services, see *Administering LiveCycle ES* at http://www.adobe.com/go/learn_lc_administration.

User management and authentication

LiveCycle Foundation includes the User Manager component, which allows administrators to maintain a database, which is synchronized with one or more third-party user directories, for all users and groups. User Manager provides authentication, authorization, and user management for services. User Manager enables Single Sign-On (SSO). When SSO is enabled, the LiveCycle ES user login pages are not required and are not displayed.

Because User Manager is built into LiveCycle Foundation, it authenticates any users who work with LiveCycle ES. User Manager implements role-based access control so that administrators can associate users and groups with roles that are already part of the User Manager database. Role-based access control requires access rights to be assigned to roles rather than to individual users. Using the User Management pages in LiveCycle Administration Console, administrators assign appropriate roles to users or groups.

Administrators can create custom roles and associate them with existing permissions. Administrators can also add new users to the database through the LiveCycle Administration Console.

LiveCycle Administration Console

The LiveCycle Administration Console gives administrators access to tools so that they can perform these tasks:

- Configure and manage users, groups, and server authentication settings using User Management.
- Create and manage invocation endpoints and deploy LCA files without the need for developer tools
- Set up watched folders and email providers for non-programmatic process invocation
- Administer solution component properties and server settings such as port numbers and log files

LiveCycle ES provides a set of tools that developers can use to define, design, and develop applications. LiveCycle Workbench ES includes robust sets of prebuilt components and other application building blocks so that developers, designers, and business analysts can work together to create engaging user interfaces and process flows.

Developing processes and forms

Using Workbench ES, you can develop forms and process flows, which are deployed to and managed by the LiveCycle ES server. Developers export the forms, processes, and related assets such as images, DDX schemas, and XML schemas as a LiveCycle ES archive (LCA) file, which can then be deployed as an application to the LiveCycle ES server.

Workbench ES includes the tools necessary for designing form guides, PDF forms, process flows, and output templates in a collaborative environment.

Designing processes

Using the LiveCycle Process Design perspective in Workbench ES, developers can create *process diagrams*, which are visual representations of the business processes that require automation. Automated processes are the electronic implementations of existing business processes.

Process diagrams can include or assemble tasks that involve any services that you have licensed. For example, you can create a process that includes these tasks:

- Retrieves a form from the LiveCycle ES repository
- Applies usage rights and rights management policies to the form
- Sends the form to a user to fill
- Returns and authenticates the form

Based on the data contained in the filled form, the process then routes the form to a different department in your organization.

Designing forms

Using the LiveCycle Form Design perspective in Workbench ES, developers can design a form, define its logic, modify it to match paper counterparts or meet strict legislative requirements, and then preview the form. Developers can use the LiveCycle Form Design perspective to create applications that generate dynamic, data-driven documents, and produce customized business documents for print, web, or archival.

Form developers can build and maintain data capture solutions that read from, validate against, and add to corporate data sources. Developers can integrate PDF documents into existing workflows by binding forms to XML schemas, XML sample files, databases, and web services.

Adobe LiveCycle Designer ES supports XML templates for forms and documents that can be merged with business data and rendered as PDF documents, HTML, and data streams for Printer Control Language (PCL), Adobe PostScript®, and Zebra Programming Language (ZPL) printers.

Business users and application developers can create form guides that automate data collection and submission by walking users through a form.

Fragments

Fragments are reusable components of a form or document. Fragments stored in a file system or in the repository are referenced by form templates created in Designer ES. Each time a template is rendered (for example, as a PDF document) at run time, the appropriate fragments are used. When designing templates that incorporate fragments, you can customize the template by using data values to select from a group of fragments.

Form guides

Designer ES integrates with Flex to provide the capability to create a wizard-like interface called a *form guide*, which is based on the Flash technology, for a form. Form guides prompt users to fill in data and move on to the next input section, making form filling easier and more intuitive. Existing forms can also be used to create form guides. Form guides are also associated with a form so that record documents are maintained as well. Form guides are an alternative view of a form that guides users through the form-filling process step by step.

Developing applications

Developers create applications using Workbench ES tools. Forms created using the LiveCycle Form Design perspective, process diagrams created using the LiveCycle Process Design perspective, or form guides created using Designer ES are packaged into an archive file. Additional resources that the process requires, such as XML metadata, images and Document Description XML (DDX) schemas, are also included in the archive. This archive file represents an application, which is deployed to the LiveCycle ES server.

Developing rich Internet applications

The Eclipse-based development environment can include Flex Builder (available separately), which developers can use to develop RIAs, that facilitate end-user participation in business processes. With Flex components developers can extend the functionality of form guides, and also let developers customize Workspace ES.

LiveCycle ES SDK

LiveCycle Foundation provides tools for programmers who plan to develop custom services to run on LiveCycle ES and client applications for invoking services.

Programmer tools include reference and task-based documentation to enable programmers to use the Java APIs, WSDLs, and LiveCycle Remoting to programmatically interact with services. Also included are complete Javadoc documentation and samples for developing custom services and applications as well as customizing the Workspace ES and form guide interfaces.

For more information about developing client applications using the APIs and LiveCycle Remoting, see the *LiveCycle ES SDK Help* at http://www.adobe.com/go/learn_lc_programming.

Adobe LiveCycle Process Management ES lets users design, manage, monitor, and optimize human-centric processes. With Process Management ES, users can also manage automated business-process applications that connect systems and people. Process Management ES enables the automation of human tasks and long-lived processes that involve asynchronous transactions.

Key features

Process Management ES offers the following features:

- Task assignment and management
- Event management
- Workspace ES

Task assignment and management

Process Management ES provides task assignment and task management services that business analysts and developers use within Workbench ES.

Business analysts and developers can use the User service within Workbench ES to define task assignments in a long-lived process. Each task assignment defines the initial user, the form type that is routed to a user, task instructions, as well as rules for reminders, escalations and deadlines.

Process Management ES supports several form types, including Acrobat forms and XDP forms rendered as PDF, HTML, and form guides. Developers can also augment the User service with data mapping.

The Task Manager service routes work (tasks) to end users through assigned queues. It coordinates tasks submitted through Workspace ES. The Task Manager service can also route Flex applications built using Flex Builder and LiveCycle Data Services ES.

Event management

Process Management ES provides the capability to create and receive business events that occur in processes. It stores events and delivers them as defined in the system through callbacks to event handlers. Events can be created or received within processes or linked to external events on a messaging bus through integration with Java Message Service (JMS).

LiveCycle ES supports several event types:

Asynchronous events: A business analyst or developer can define asynchronous events such as Cancel Order or New Order that can be linked to processes either to enable the initiation of a process or to handle a complex process flow within an existing process. A process could be initiated based on a New Order event and, while it is running, could receive a Cancel Order event that enables it to change its flow of execution based on its state. If the order was not fulfilled, the process could refund the customer or, if the order was shipped, the process could send an invoice to the customer.

Exception events: These events are usually defined by component developers and allow handling of errors during process execution. For example, the server is not available and triggers an alert to an administrator, or a transaction failure allows the developer to define a compensating transaction.

Timer events: These events allow processes to wait for a period of time or can be associated with activities so that a different process flow is executed if they are not completed on time.

Event filters: These events allow processes to look for events that contain specific data. For example, a process can send a document for signature to a customer and then wait for an event when the customer returns a signed document. The event filter can filter the event based on the process ID or customer reference.

LiveCycle Workspace ES

LiveCycle Workspace ES is an intuitive Flex-based application that lets end users initiate and participate in form-based business processes by using a web browser.

Using Workspace ES, users can perform these tasks:

- Fill forms that initiate a business process
- Open forms that have been routed to the user or to a group queue to review, approve, or provide more information
- Add attachments and comments to a task and restrict access
- Search for forms that are part of a completed business process or active processes that the user has initiated or participated in
- Have custom searches and filters based on process variables
- View process categories and a list of tasks
- Select processes and place them in a Favorites folder for easy access
- Share tasks and consult with other users
- Track processes and look at the audit trail.

In the development environment, developers can use Flex Builder to customize the Workspace ES web interface that end users see. This allows your organization to ensure that the Workspace ES user interface is branded appropriately and customized to meet your business requirements.

Workspace ES also integrates with LiveCycle Data Services ES to automatically synchronize data with the LiveCycle ES server. (See ["LiveCycle Data Services ES" on page 25](#).)

Services included with Process Management ES

Process Management ES includes these services:

- Encryption
- Form Augmenter
- Form Data Integration
- PDF Utilities
- Stall
- Task Manager
- User
- Wait Point
- Workspace
- XMP Utilities

For more information about the services included in this solution component, see
http://www.adobe.com/go/learn_lc_services.

Adobe LiveCycle Business Activity Monitoring ES is a collection of client applications that run on Microsoft Internet Explorer browsers and connect to servers that collect, store, and aggregate event and context information. Business Activity Monitoring ES stores information about the run-time objects in a metadata database, sends email alert notifications through an SMTP mail manager, and receives and retrieves event and context data through agents. *Agents* are processes that convert external data to a Business Activity Monitoring ES format.

Business data modeling is a technique for describing the events, context, and rules that depict how your business functions. You design a model in Business Activity Monitoring ES by combining event streams and context sources into business *views*, which are models that provide a picture of a business activity. As new events enter the system, the views are automatically updated immediately to reflect the current details about the activity.

In addition to the views, another part of modeling is the ability to create and test scenarios. Using scenarios, you can test for expected or possible outcomes and identify exceptional business conditions. Each scenario contains rules that identify specific possible conditions, and alerts and reportlets to send to key personnel when the condition's existence is verified.

Key features

Business Activity Monitoring ES offers these features:

- Analytics server
- Performance dashboards
- Analytical workbench

Analytics server

The analytics server provides these capabilities:

- Is based on event-driven processing, which captures changes to systems (events and data) as they occur
- Provides real-time event and data integration, aggregating and correlating operational and historical data from multiple data sources
- Streams data stores to ensure continuous data integration
- Provides an engine for multi-dimensional analysis, dynamic modeling, business-rules execution, and exception and alert reporting
- Supports memory resident data storage and processing
- Handles a high transaction throughput and a large number of dashboard users
- Permits complex rules creation and quick updates, as well as comprehensive exception and alert reporting.

Performance dashboards

Performance dashboards provide visibility into all of your critical business information. BAM Dashboard is an intuitive interface that you can use to customize metrics, set alerts, and drill down to detailed data. The interface also provides a robust graphical object library that includes process diagrams, geographic maps, indicators, charts, and tables.

BAM Dashboard can be set to role-based, which provides operational visibility by job title or function, personalization by role, function or individual with centralized control, automatic data filtering, and role-based security. This allows Process Management ES to forward important alerts or metrics to the appropriate individual, based on the individual's job title or function within the organization.

Analytical workbench

BAM Workbench allows system administrators to set up data integration, analytic models and end-user dashboards to show current corporate information, historical data or aggregated views.

System administrators set up data connectivity by configuring event and contextual data sources, and accessing multiple, concurrent data streams. They can also create business rules, dashboard objects, as well as views and cubes.

Adobe LiveCycle Data Services ES provides a comprehensive set of data-enabling features, which are deployed in a Java web application. Data Services ES also provides a separate set of features specifically for working with data in LiveCycle Foundation. These features are installed and deployed in LiveCycle Foundation.

Data Services ES lets you build data-rich Flex and Ajax applications that securely and cooperatively interact with a variety of data sources, including web services, XML data, and custom Java destinations, including the Data Management Service, Hibernate, and SQL.

Key features

Data Services ES provides the following set of services that let you securely access, transmit, display, and update data:

Message Service: Use a client-side API and a corresponding server-side Message Service to create Flex messaging applications.

RPC services: Access data through HTTP GET or POST (HTTP services), SOAP (web services), or Java objects (remote object services).

Data Management Service: Build applications that provide data synchronization, data replication, and occasionally connected application services. The Data Management Service provides support for paged data, which lets you manage large result sets by returning subsets of data, called *pages*, to the Flex client application.

Conflict resolution: Data Management Service tracks the original state of the object so that it can accurately detect both types of conflicts and provides an API for programmers to use to resolve them.

PDF creation: Build a Flex application that can generate a PDF document that includes static images and/or dynamic or static data.

Security: Restrict access to a privileged group of users by applying a security constraint in a destination definition.

Flex-Ajax Bridge: Expose a Flex application or component to scripting in a web browser.

Ajax Data Services: Lets Ajax developers access the messaging and data management capabilities of Data Services ES directly from JavaScript™.

Proxy: Redirects service traffic from different domains. Provides client authentication, whitelists of permitted Remote Procedure Call (RPC) service URLs, server-side logging, localization support, and centralized management of RPC services.

WSRP generation: Lets you deploy a Flex application as a portlet in a portal server without having to do any portal-specific programming.

Offline data caching: Lets you cache client data requests and data changes to the local file system for later retrieval when an application resumes.

Realtime Quality of Service: Lets Flex clients select custom, data-access policies for real-time data.

Open adapter architecture: The Data Management Services lets you retrieve data by using a variety of mechanisms. Data Services ES includes adapters for JMS, Hibernate, Java, SQL, and ActionScript™.

Automated testing: Create applications and components that can be tested with Mercury QuickTest Professional (QTP).

JSP tag library: Lets you embed MXML and ActionScript code into a JavaServer Pages™ (JSP), providing an easier entry for J2EE developers to Flex programming.

The following features are integrated with LiveCycle Foundation:

LiveCycle Remoting: Invoke services through the Flex RemoteObject tag.

Proxy: Redirects service traffic from different domains. Provides client authentication, whitelists of permitted URLs, server-side logging, localization support, and centralized management of LiveCycle Remoting.

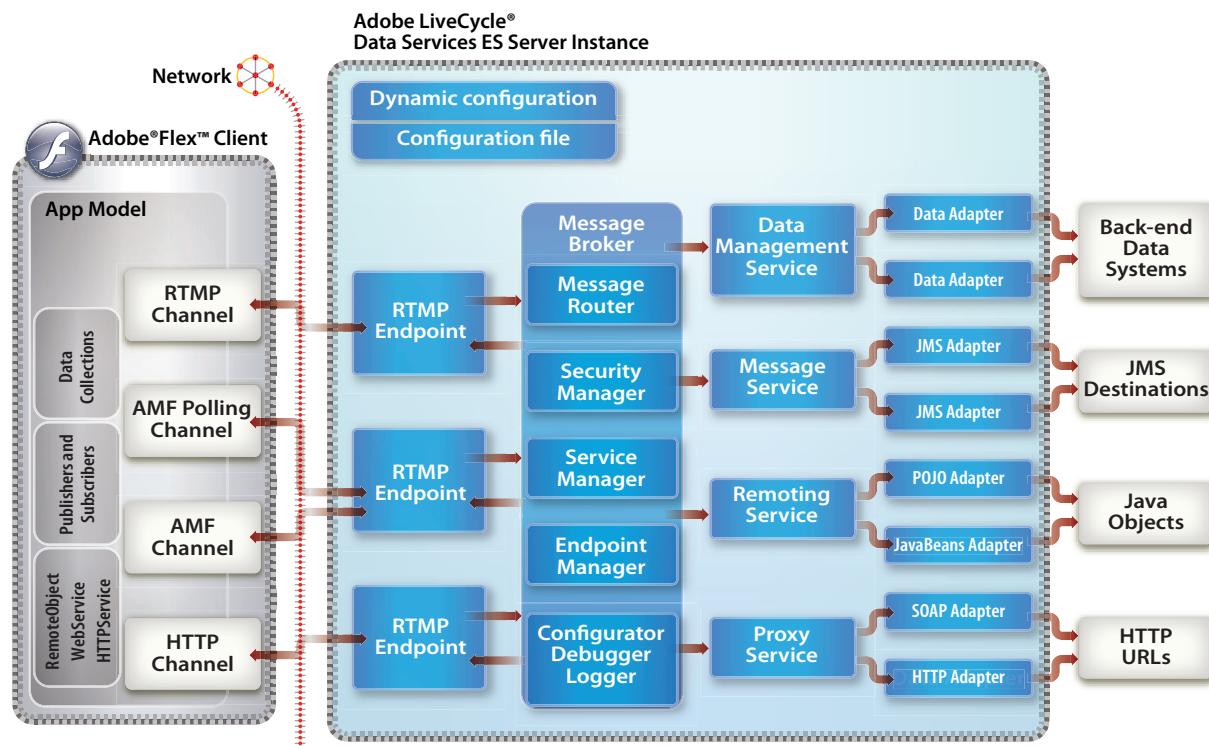
How Data Services ES works

Data Services ES is a J2EE web application that you can deploy on a variety of J2EE application servers and servlet containers.

The Data Management Service, the Message Service, and the Remove Procedure Call (RPC) services are all built on a core messaging infrastructure. Data Services ES uses the messaging infrastructure and the Action Message Format (AMF) protocol to ensure optimum performance. You can also use other protocols, such as HTTP and Real Time Messaging Protocol (RTMP).

Data Services ES uses XML configuration files to manage channels, destinations, message topics, message queues, and other settings. Data Services ES also provides run-time configuration. Server configuration can also be defined programmatically, which improves the ease of integrating existing code into a Data Services ES application.

The Flex framework provides MXML and ActionScript APIs that let you use Data Services ES in your applications. With the Data Management Service, multiple clients are automatically synchronized when they retrieve data from the same destination. With the Message Service, you can create applications that act as producers, consumers, or both. Data Services ES client applications use channels and destinations that are declared on the Data Services ES server instance. They can also use dynamic configuration to create destinations at run time.



Using Data Services ES

With Data Services ES, developers define a set of *destinations* using XML configuration files. These definitions are used by the built-in service adapters provided as part of the application. These include low-level adapters to connect to Java objects (data access objects), Java Message Service (JMS) topics and queues, or Adobe ColdFusion® components (CFCs) as well as higher-level adapters for common persistence solutions such as Hibernate, Enterprise JavaBeans (EJB), and Spring.

The following sections highlight major services in Data Services ES.

For complete information, see *LiveCycle Data Services ES Developer's Guide* at www.adobe.com/go/learn_lc_dataServicesDevGuide.

LiveCycle Remoting

LiveCycle Remoting integration lets Flex clients call services in LiveCycle Foundation. It supports synchronous and asynchronous service invocation. LiveCycle Remoting performs much faster than web services by using the more efficient AMF3 binary protocol.

Message Service

This messaging feature is based on established publish and subscribe messaging standards and terminology. It provides a client-side API and a corresponding server-side Message Service for creating Flex messaging applications. The Message Service also enables participation in JMS messaging. The messaging and real-time infrastructure enables collaboration and data-push applications to be built in a scalable and reliable manner while preserving the lightweight web deployment model.

Data Management Service

The Data Services ES data management feature spans the client and server tiers to provide the top-level functionality for distributed data in Flex applications. This feature lets you build applications that provide data synchronization, data replication, and occasionally connected application services. Additionally, you can manage large collections of data and nested data relationships, such as one-to-one and one-to-many relationships, and use Data Management Service adapters to integrate with data resources.

A client-side Data Services ES component, which you can create in MXML or ActionScript, calls methods on a destination configured in the server-side Data Management Service. The client-side component performs activities such as filling client-side objects with data from remote data resources and synchronizing the versions of data in multiple client instances with the server-side destination.

RPC services

Remote Procedure Call (RPC) services let you interact with server-side RPC services to provide data to your applications.

You can access data through HTTP GET or POST (HTTP services), SOAP (web services), or Java objects (remote object services). Another common name for an HTTP service is a REST-style web service. REST stands for Representational State Transfer and is an architectural style for distributed hypermedia systems.

In a typical Flex application, an RPC component sends data as input to one or more RPC services. When an RPC service executes, it returns its results data to the RPC component that made the request.

LiveCycle Remoting is an extension of the Remoting RPC service.

Flex-Ajax Bridge

The Flex- Ajax Bridge (FABridge) is a small code library that you can insert into a Flex application, a Flex component, or even an empty SWF file to expose it to scripting in the browser.

Instead of defining new, simplified APIs to expose a graph of ActionScript objects to JavaScript, you can use FABridge to make your ActionScript classes available to JavaScript without any additional coding. After you insert the library, you can do anything with JavaScript that you can do with ActionScript.

Ajax Data Services

Ajax Data Services is a JavaScript library that lets Ajax developers access the messaging and data management capabilities of Data Services ES directly from JavaScript. It lets you use Flex clients and Ajax clients that share data in the same messaging application or distributed data application.

Many Ajax applications are taking on the responsibilities of round-tripping data. Using Ajax Data Services, you can integrate publish subscribe messaging, real-time streaming data, and a rich data management API that dramatically simplifies the development of rich Internet applications (RIAs).

RIA-to-PDF generation

Data Services ES provide the ability for Flex applications to generate PDF documents that include graphical assets created using Flex or Flash. The generated PDF document can be coordinated with other services. For example, a PDF document that is generated from a Flex 2 application using Data Services ES is assembled together with other PDF documents into a single PDF document. Then it can be secured with a policy by using Rights Management ES and emailed to a customer as part of a process coordination.

Adobe LiveCycle Forms ES enables organizations to extend their intelligent data capture processes by deploying electronic forms in Adobe PDF, HTML, and SWF format over the Internet. Forms ES also retrieves form data from central repositories and merges it with the specified form. With Forms ES, end users can access online forms without downloading any additional software, fill the forms online, or save them to fill offline.

Key features

Forms ES offers the following key features:

- Renders PDF, HTML, or SWF form guides
- Enables form data integration, which is the ability to import data into and extract data from PDF forms
- Includes Data Services ES support for form guide rendering
- Renders forms based on fragments
- Performs form assembly

Rendering forms and integrating form data

Forms ES lets you render and process interactive forms and large data sets. You can create applications that perform interactive form rendering operations such as these:

- Render interactive PDF documents. For forms created in Designer ES that have a flowable layout, Forms ES adds extra fields and boilerplate as a result of merging the form design with data or as a result of scripting.
- Render interactive HTML forms. Forms ES automatically detects the browser type and platform, and then dynamically generates an HTML document that is based on a form design created in Designer ES.
- Detect whether form design scripts should run on the client or on the server when rendering the form.
- Validate data entry by performing calculations, accessing databases, or enforcing business rules on field-level data, and then return the resulting data to the browser.
- Load XML data into an XML Data Package (XDP) file or into a PDF file that contains XDP information. Forms ES retrieves form data from central repositories and merges it with the specified form when rendering the form.
- Extract XML data from an XDP file. Forms ES can process form data submitted from a user. Form data can be submitted to an organization's core systems, thereby increasing the quality of data gathered, improving customer service, and leveraging investment in core systems.

Creating forms using form guides

With form guides, you can create a more engaging form filling experience by streamlining and simplifying the way in which a form filler inputs form data.

New form guides are created using Guide Builder, which is based on the Flash technology and is included in the LiveCycle Form Design perspective of Workbench ES. Using Guide Builder, users can build form guides based on an existing form design.

After users create a form guide, they must render it using either the Forms ES API or the Render Form service designed in Workbench ES (or a custom service) before the form guide can be made available to end users. Customers can also take advantage of Flex Builder to create custom form guide components based on the form guide components that are included with the LiveCycle Form Design perspective in Workbench ES.

Rendering forms based on fragments

Forms ES can render forms that are based on fragments that you create using Designer ES. A *fragment* is a reusable part of a form and is saved as a separate XDP file that can be inserted into multiple form designs. For example, a fragment could include an address block or legal text.

Using fragments simplifies and speeds up the creation and maintenance of large numbers of forms. When creating a new form, you insert a reference to the required fragment and the fragment appears in the form. The fragment reference contains a subform that points to the physical XDP file.

Here are advantages of using fragments:

Content reuse: You can reuse content in multiple form designs. When you need to use some of the same content in multiple forms, it is faster and simpler to use a fragment than to copy or re-create the content. Using fragments also ensures that the frequently used parts of a form design have consistent content and appearance in all the referencing forms.

Global updates: You can make global changes to multiple forms only once, in one file. You can change the content, script objects, data bindings, layout, or styles in a fragment, and all XDP forms that reference the fragment will reflect the changes.

Shared form creation: You can share the creation of forms among several resources. Form developers with expertise in scripting or other advanced features of Designer ES can develop and share fragments that take advantage of scripting and dynamic properties. Form designers can use those fragments to lay out form designs and to ensure that all parts of a form have a consistent appearance and functionality across multiple forms designed by multiple people.

Assembling PDF documents and forms

Forms ES can combine content from multiple PDF documents to create consistently formatted and seamless PDF documents.

You can use Forms ES for document assembly in the following types of workflows:

Customer communications: Automating the batch creation and assembly of customer invoices, statements, letters, form packages, insurance policy statements, marketing materials, and loan packages

Regulatory filings: Integrating document assembly into a regulatory document filing process

Archive preparation: Automating batch document customization for archiving (such as adding watermarks, or inserting or extracting metadata)

Sales force automation: Preparing requests for quotes (RFQs) or generating proposals from multiple sources

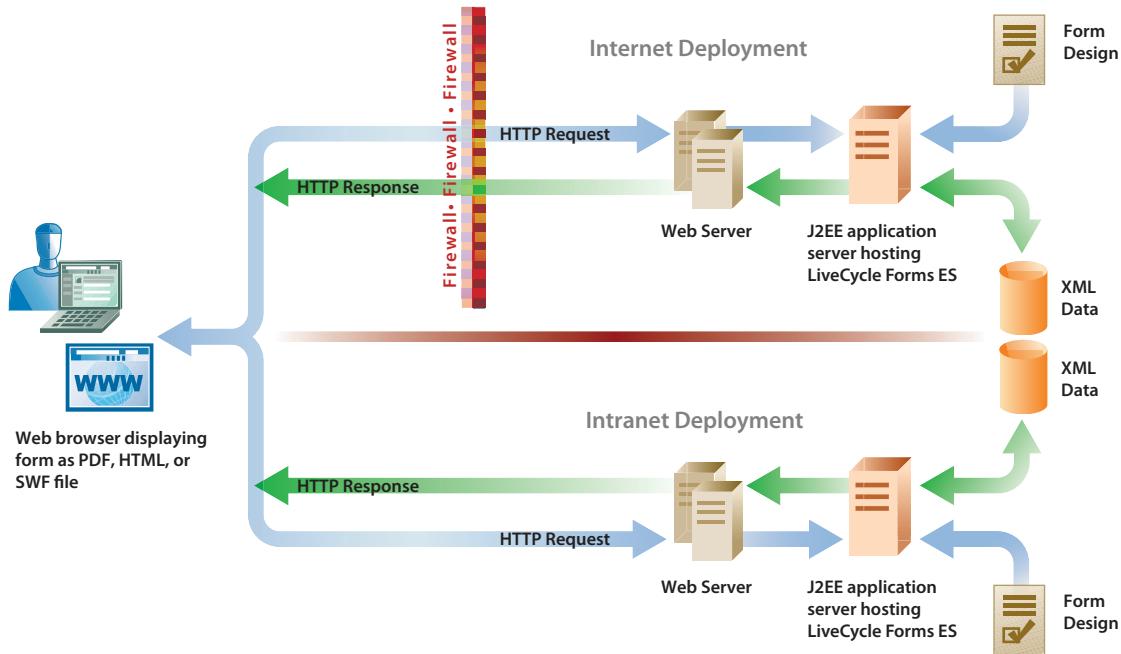
How Forms ES works

The forms for use with Forms ES are typically created in Designer ES. Forms ES also supports Acrobat forms. The form author can deploy the form designs for use with Forms ES either as XDP files or as PDF files, depending on the requirements of the business process. Forms ES can render an XDP file as an HTML form, a PDF form, or a form guide. A PDF file is rendered as a PDF form.

The end-user environment consists of a web browser (for HTML forms or forms that use the Flash technology) together with Adobe Reader (for PDF forms). Forms ES can detect the browser type and dynamically generate a PDF form, an HTML form, or Flash content based on the form design created in Designer ES and the form preference that the developer identifies in the LiveCycle ES application.

When end users request a document from Forms ES (for example, by clicking a button or an image on a web page), the request initiates a series of specific processes and interactions among the web application, Forms ES, and the web browser. After receiving the form, end users can interact with it online. After end users are finished with the form, they submit it, along with form data, back to Forms ES.

The following illustration and tasks provide an example of how Forms ES processes a request from an end user.



1. The end user accesses a web page and requests a form.
2. The web application invokes Forms ES and requests the form.
3. Forms ES retrieves the form design from a repository and data (possibly from an enterprise database), and then merges the form design with the data to prepopulate parts of the form. The data can come from a variety of sources, such as an enterprise database, another form, or another application.
4. Forms ES determines the format in which to render the prepopulated form based on the browser information that is passed with the call. The format of a form can also be set programmatically by using the Forms service API.

5. Forms ES transforms the form design into PDF, HTML, or Flash content and then returns the prepopulated form to the end user.
6. The end user completes the form and then submits the form data back to Forms ES. Before form data is submitted back to Forms ES, applicable client-side scripts are executed. For example, a user may be prompted to provide a value for a mandatory form field.
7. Forms ES extracts the submitted data, runs any server-side scripts associated with the button that was clicked, and then executes the calculations and validations on the form.
8. Forms ES returns the results. If validations fail, the result may be a form that is returned to the end user. However, if validations are successful, the result may be XML data.

Services included with Forms ES

Forms ES includes these LiveCycle ES services:

- Forms
- Assembler
- Encryption
- Form Data Integration
- PDF Utilities
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

Adobe LiveCycle Output ES provides support for a variety of output formats and supports the output design features provided by the LiveCycle Form Design perspective in LiveCycle Workbench ES or LiveCycle Designer ES.

Key features

Using Output ES, you can create applications with the following functionality:

- Generate final form documents by populating Designer ES files with XML data
- Output forms in a range of formats including non-interactive PDF, PostScript, PCL, and ZPL print streams
- Control printer features, select paper trays, duplex and staple printed documents
- Process single documents, document packages, and document batches from single or multiple Designer ES templates that may use fragments
- Assemble PDF documents and convert PDF documents from native documents
- Convert PDF packages that may contain any combination of interactive PDF documents and non-interactive PDF documents to PostScript

Document design for Output ES

Designer ES form design features include tools for designing fixed and flowable document layouts for Output ES. The layout and output choices you make when you design forms and documents are implemented and performed by Output ES.

Using Designer ES, form authors can use a variety of layout specifications and features in form designs and templates:

- Create forms whose layouts adjust when merged with XML data. This ensures that rendered (or printed) forms accommodate the merged data content and volume by including or excluding design elements, growing to create space for data, and paginating automatically.
- Define duplex formatting and the media (paper tray) intent in a form template to logically link document layout to printer capabilities.
- Access fragments in the repository and incorporate them into form templates for greater control and management of document designs.
- Spell-check documents and generate sample test data while designing forms. Using the sample test data, you can then generate a PDF preview or print directly to a target printer.

These features were designed for use with Output ES. By incorporating the layout and print control features into a business process, forms and documents created with these layout features can be rendered or printed in a variety of formats without compromising the design intent.

Supported document formats

Using Output ES, processes can merge XML data with forms or documents created using Designer ES to produce documents in a variety of formats, including the following types:

- PDF final form documents for viewing and printing in Adobe Reader or Acrobat
- PDF archive documents in PDF/A (1A and 1B) or ISO standard
- PostScript and PCL documents for direct printing to monochrome and color laser printers
- ZPL for direct printing of documents to Zebra label printers

Output ES also lets you generate documents in a wide range of languages, including Western European, Eastern European, Asian, and Middle Eastern character sets.

Controlling print features

Output ES provides the ability to leverage and control the specific capabilities of the printers you use to generate printed documents. Using the Device Profile Editor, you can edit existing device profiles (XDC files), create new ones, and map media to specific paper trays.

You can generate documents that access specific features of laser printers, including duplex printing, specifying which input and output paper trays to use, the use of resident fonts, as well as stapling and copying.

You can generate PDF documents that automatically open a print dialog box in Adobe Reader or Acrobat, or that are preconfigured to specify duplex printing and the number of copies to be printed on a default or specified printer.

Generating flexible documents

Output ES offers flexible document generation capabilities. You can generate single documents, each from specific Designer ES templates; a batch of documents from a single template; or a batch of documents, each from separate files by using a single Designer ES template.

Using this flexible generation capability, you can also leverage fragments in document output. You can generate a document package that uses multiple Designer ES templates. For example, a mortgage application can contain numerous sections, some of which are customized for the client by using specific templates for interactive forms and fragments, and some of which are standardized by using dynamic PDF documents. The final generated output is a single cohesive document.

With Output ES, you can also produce companion metadata files for document integration and statistics. For example, metadata from incoming XML such as submitted fax numbers or account numbers can be integrated into forms for output. Output ES also supports metadata that is used for generating documents, such as page numbering and the specific documents included in a batch.

Document conversion and assembly

Output ES supports document generation beyond simple print control and PDF rendering. It also supports PDF document assembly and conversion. For example, you can convert PDF documents to TIFF files for archiving or to PostScript files for direct server-based printing. You can convert interactive PDF forms (fillable forms) to final version PDF forms (filled information is frozen, and the document is no longer interactive).

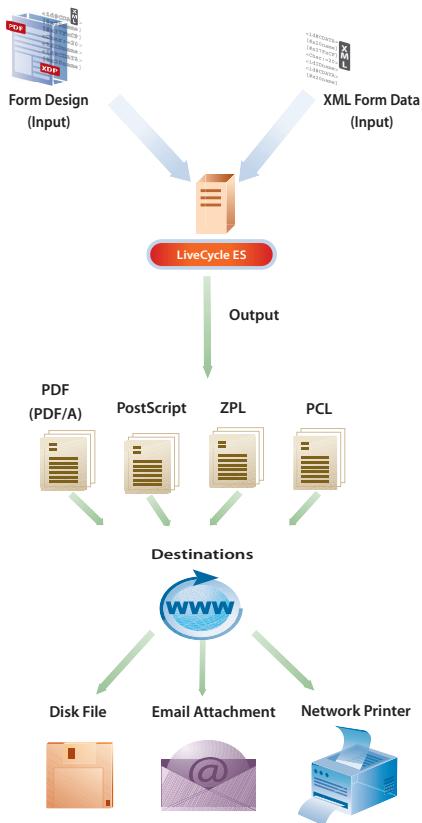
If you have multiple, disparate PDF forms that you need to merge into a single document, you can use Output ES to manipulate and assemble the multiple source PDFs into a single seamless document.

Document and process automation

You can develop processes that automate generating documents that include barcodes containing variable data or that print directly to industry-standard Zebra (ZPL) label printers. Barcode data can be a range of standard codes, including two-dimensional symbologies such as PDF417, Datamatrix, or QR. When you output ZPL content to a label printer, you can use standard or RFID labels, and access Zebra barcodes and standard Zebra fonts.

How Output ES works

Here is an example of how Output ES works.



A form author creates the form in Designer ES. The form conforms to the XML schema that the Output service will use as the XML data input file to create the output.

1. The developer identifies the form, XML data input file, and output format that the Output service will use to create the output. The developer uses one of these methods to identify the form, input, and output:
 - Adds the Output service in a process created in Workbench ES
 - Uses the APIs provided with the LiveCycle ES SDK
2. The developer deploys the application to the LiveCycle ES server.

3. The Output service is invoked. An XML input file is provided to the Output service. The service extracts the form design from the repository and merges the XML data input file with the form design to create the output. The output is either a print stream (PostScript, PCL, or ZPL) or a PDF/A document.
4. The output is sent to the designated destination as an email attachment (when using the Email service as part of a process), sent to a network printer, or is saved to a file on disk.

Services included with Output ES

Output ES includes these LiveCycle ES services:

- Output
- Assembler
- Convert PDF
- Encryption
- PDF Utilities
- Print PDF Package
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

With Adobe LiveCycle Reader Extensions ES, your organization can extend the functionality of Adobe Reader by adding usage rights to the PDF documents it distributes. Reader Extensions ES enables your organization to easily share interactive PDF documents with external parties by extending the functionality of Adobe Reader. Reader Extensions ES supports any PDF document rendered in Adobe Reader 7.0 and later.

Usage rights are permissions that Reader Extensions ES adds to a PDF document to activate features that are not usually available when a PDF document is opened using Adobe Reader. Adobe Reader users do not require additional software or plug-ins to work with PDF documents enabled by Reader Extensions ES.

Reader Extensions ES lets you select usage rights for individual or batches of PDF documents by using the wizard-like, web-based application. Using the developer or programmer tools, you can assign usage rights to a large number of documents in an automated batch process.

Key features

PDF documents that have usage rights added let document recipients and users do the following activities:

- Complete PDF documents and forms online or offline, allowing recipients to save copies locally for their records and still keep added information intact
- Save PDF documents to a local hard drive to retain the original document and any additional comments, data, or attachments
- Attach files and media clips to PDF documents
- Sign, certify, and authenticate PDF documents by applying digital signatures using industry-standard public key infrastructure (PKI) technologies
- Submit completed or annotated PDF documents electronically
- Use PDF documents and forms as an intuitive development front end to internal databases and web services
- Share PDF documents with others so that reviewers can add comments by using intuitive markup tools, such as electronic sticky notes, stamps, highlights, and text strikethrough (the same functions that are available in Acrobat)
- Supports barcoded forms decoding when installed with LiveCycle Barcoded Forms ES licensing

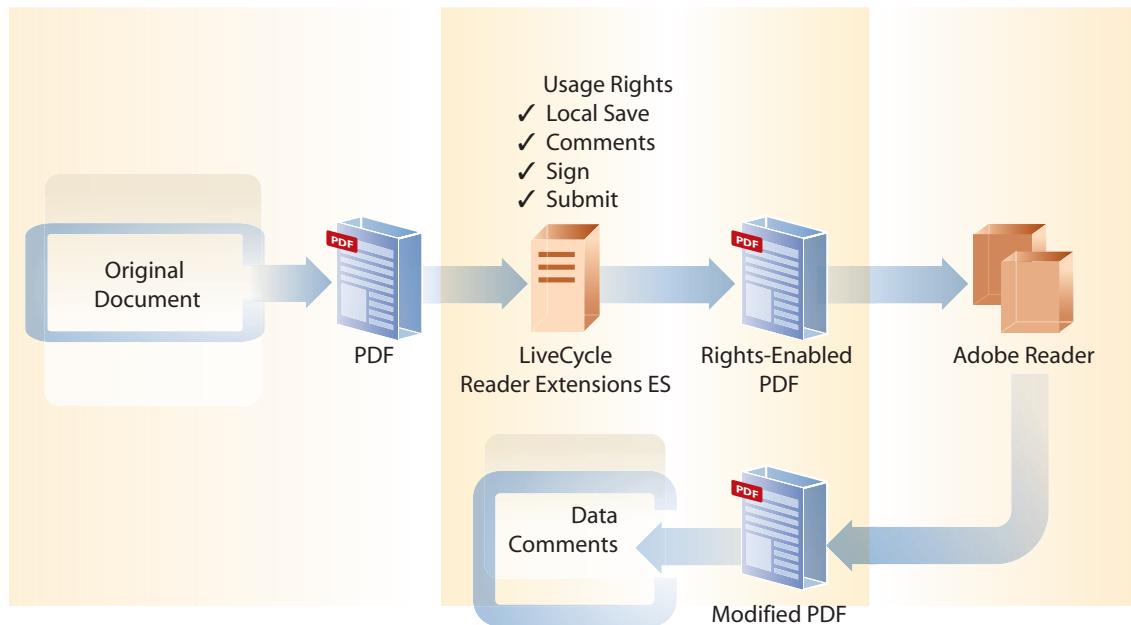
These special user capabilities are automatically activated when a rights-enabled PDF document is opened within Adobe Reader. When the user is finished working with a rights-enabled document, those functions are once again disabled in Adobe Reader until the user receives another rights-enabled PDF document.

The specific usage rights that you can set are determined by a credential that you purchase from Adobe. Credentials typically provide permission to set a group of related usage rights, such as those pertaining to interactive forms. Each credential provides the right to create a certain number of rights-enabled documents. An evaluation credential provides permission to create an unlimited number of draft documents within a limited time period.

How Reader Extensions ES works

Most of the complexity of Reader Extensions ES is hidden from anyone who interacts with the rights-enabled PDF documents or selects usage rights through the web-based application. However, before you implement LiveCycle ES, it is important to understand how it works.

Here is an example of how Reader Extensions ES works.



1. A developer selects the usage rights by using any of these methods and uploads the document to the LiveCycle ES server:
 - Accesses the web-based application provided with Reader Extensions ES
 - Adds the Reader Extensions service in a process created in Reader Extensions ES
 - Uses the APIs provided with the LiveCycle ES SDK
2. Reader Extensions ES adds the selected usage rights to the document.
3. The rights-enabled PDF document is available.
4. When end users open the rights-enabled document using Adobe Reader, they can interact with the document according to the usage rights that are added to it.
5. End users can sign and submit the document electronically or forward the document to others who can interact with the document according to the embedded usage rights.
6. If the PDF document is returned to the originating enterprise, the enterprise applications extract the updated information.

Services included with Reader Extensions ES

Reader Extensions ES includes these services:

- Reader Extensions
- Encryption
- Form Data Integration
- PDF Utilities
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

With Adobe LiveCycle Barcoded Forms ES, you can add one-dimensional and two-dimensional barcodes to interactive PDF forms. You can then publish the barcoded forms to a website or distribute them by email or CD. When a user fills a barcoded form by using Adobe Reader or Acrobat, the barcode is updated automatically to encode the user-supplied form data. The user can submit the form electronically or print it to paper and submit it by mail or fax. You can later extract the user-supplied data as part of a LiveCycle ES process, routing the data to the appropriate business processes based on the form type or the data itself.

Barcoded forms can eliminate the need for optical character recognition (OCR)-based forms processing and the attendant costs of manual data entry. Data captured from fill-and-print barcoded forms can be reinserted into your electronic process quickly and automatically with 100% accuracy. Furthermore, you can retain a digital image of the submitted signed form for archiving purposes.

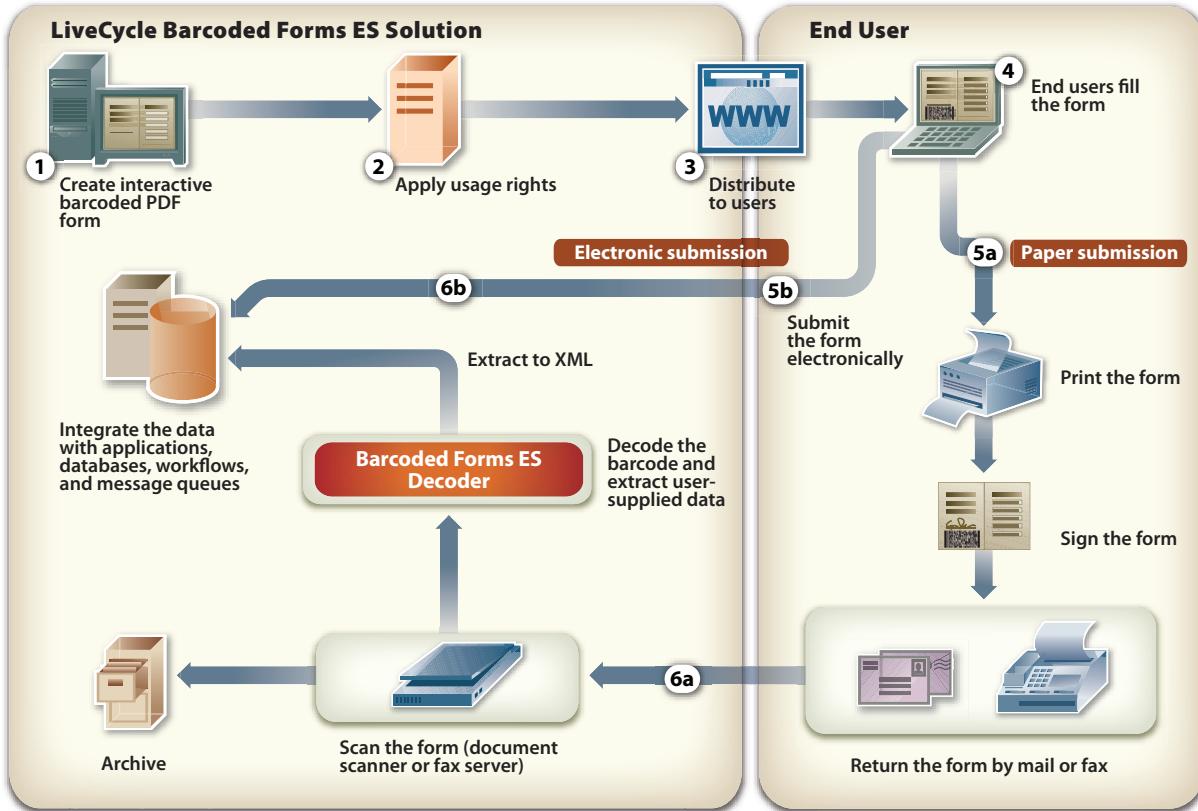
Key features

Barcoded Forms ES offers the following key features:

- Provides a unified approach for both paper and digital forms processing
- Automates the extraction and translation of barcoded data into core IT processes
- Supports barcodes encoded as XML, tab-delimited, or other user-defined formats

How Barcoded Forms ES works

Here is an example of how Barcoded Forms ES works.



1. Your organization's form author creates an interactive barcoded PDF form using Designer ES or Acrobat Professional.
2. Using the Reader Extensions ES web application, a user applies usage rights to the barcoded PDF form.
3. The user electronically publishes the barcoded form using the web, email, or a CD.
4. The end user opens the barcoded PDF form in Adobe Reader or Acrobat and fills the form. As the user fills the form, the user's data is automatically encoded in the barcode.
5. a) For a paper submission, the user prints and signs the form, and mails or faxes the form to your organization.
b) For an electronic submission, the user clicks a submit button to submit the form data electronically.
6. a) For a paper submission, when the completed form is received, your organization scans the form into an electronic image. Barcoded Forms ES Decoder locates the barcode on the scanned image, decodes it, and extracts the data in your specified format.
b) For an electronic submission through the Submit by Email button, the data, other than the barcode data, is directly submitted to the processing centre as XML.

Note: Barcoded Forms ES Decoder can decode a PDF file that was saved in Acrobat when the file is directly submitted to the decoder in the same way a scanned TIFF file is submitted.

Authoring barcoded forms

Form authors create the forms using Designer ES or Acrobat Professional. In the authoring phase, the form author can specify any format to encode the data in the barcode, such as XML or tab-delimited characters.

LiveCycle Designer ES

In Designer ES, form authors create an interactive PDF form from scratch or by using a form template. Form authors can drag images and other objects, such as list boxes, text fields, command buttons, and barcodes onto the form and then resize and position them to suit your organization's requirements.

Designer ES provides more advanced features that let form authors use scripting objects, integrate a form with a data source, and create dynamic forms. One advantage of authoring forms using Designer ES is that form authors are working directly in the form's source.

Acrobat Professional

If your form authors have used Acrobat Professional to author a large number of forms that have many custom scripts attached to the form objects, they can save time and effort if they add barcodes to the forms by using Acrobat Professional.

Creating a process

Developers can optionally create a process by using Workbench ES to include business processes specific to Barcoded Forms ES. When integrated with other solution components by using processes designed in Workbench ES, a single unified forms process can easily support different paper form submissions, each with their own specific workflow.

For more information, see *LiveCycle Workbench ES Help*, which is available from within Workbench ES.

Adding barcoded forms usage rights for Adobe Reader

Your organization must add barcoded forms usage rights to a PDF document before publishing the form to your customers. These usage rights activate the barcode data so that any commercial barcode decoder can read the barcode on the PDF form. Without extending usage rights of the PDF form with the barcoded forms usage right, the barcode will be illegible to all decoders, including those provided by Adobe.

In addition to the barcoded forms usage rights, the following functionality is enabled on the form:

- Saving completed or partially completed forms locally for offline filing and archiving
- Adding comments to and routing forms through email for third-party reviews
- Applying digital signatures to authorize applications or transactions
- Submitting form data electronically

Adobe Reader 7.0 and later does not require additional software or plug-ins to work with PDF documents enabled by Reader Extensions ES.

These special user capabilities are automatically activated when a rights-enabled PDF document is opened within Adobe Reader. When the user is finished working with a rights-enabled document, those functions are once again disabled in Adobe Reader until the user receives another rights-enabled PDF document.

Usage rights are granted on a per-form basis or a per-document basis and do not apply to any other form or document. Adobe licenses the barcoded forms usage right based on the number of consumers of the form.

Updating barcodes during form completion

When a user fills the barcoded form electronically using Adobe Reader or Acrobat, the barcode is updated with the user-supplied information as the user moves between fields or saves the form.

Note: If a user fills a barcoded form by using an earlier version of Acrobat or Adobe Reader that does not support barcodes, the barcode is replaced by a gray rectangle. A gray rectangle indicates that the barcode cannot be updated and ensures that form processors do not process barcoded forms inadvertently in situations where the barcode does not accurately reflect the form's user-supplied data.

Decoding barcodes to extract barcode data

The process at your forms processing center can affect your ability to successfully process and decode barcodes from barcoded forms. Key steps in processing barcoded forms include preparing documents, capturing data from barcodes, and routing captured data to enterprise systems.

The process of capturing data from a barcode varies depending on the type of device you use to process a barcoded form. You can select from the following options:

- Document scanner and the Barcoded Forms ES Decoder
- Fax server and the Barcoded Forms ES Decoder

The Barcoded Forms ES Decoder locates the barcode on the scanned image (in TIFF or PDF), decodes it, and extracts the data in the specified format. The extracted data can then be used by another solution component such as Forms ES as part of a business process. For example, Forms ES can regenerate the original form automatically with the data entered by the user or import the data into a blank form. This completes the digital, to paper, to digital cycle (round-trip).

Processing captured barcode data

Using the process that you created, LiveCycle ES can automatically forward captured form data to the appropriate enterprise processing application. Because you can specify the data format in the authoring phase, moving form-based data across multiple enterprise applications is effortless. You can also archive data to be visually presented months or years later, exactly as it was entered into the original PDF form.

Services included with Barcoded Forms ES

Barcoded Forms ES includes these services:

- Barcoded Forms
- Form Data Integration
- Encryption
- PDF Utilities
- Reader Extensions
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

Adobe LiveCycle Digital Signatures ES lets you use digital signatures to preserve the integrity and authenticity of a document as it is transferred among users within and beyond the firewall, when it is downloaded offline, and when it is submitted back to your organization.

With Digital Signatures ES, you can automate the process of certifying and signing documents, as well as validating signatures in documents that are submitted back to your organization.

Key features

Digital Signatures ES can apply security features to any PDF document whether it is generated by other Adobe server products, on a desktop by Acrobat, or even by a third-party solution. Because PDF documents can contain any type of information, such as text, audio, and video files, you can use Digital Signatures ES to secure any type of information that is saved in a PDF document.

Digital Signatures ES can apply the appropriate security features through automated business processes or programmatically through the API:

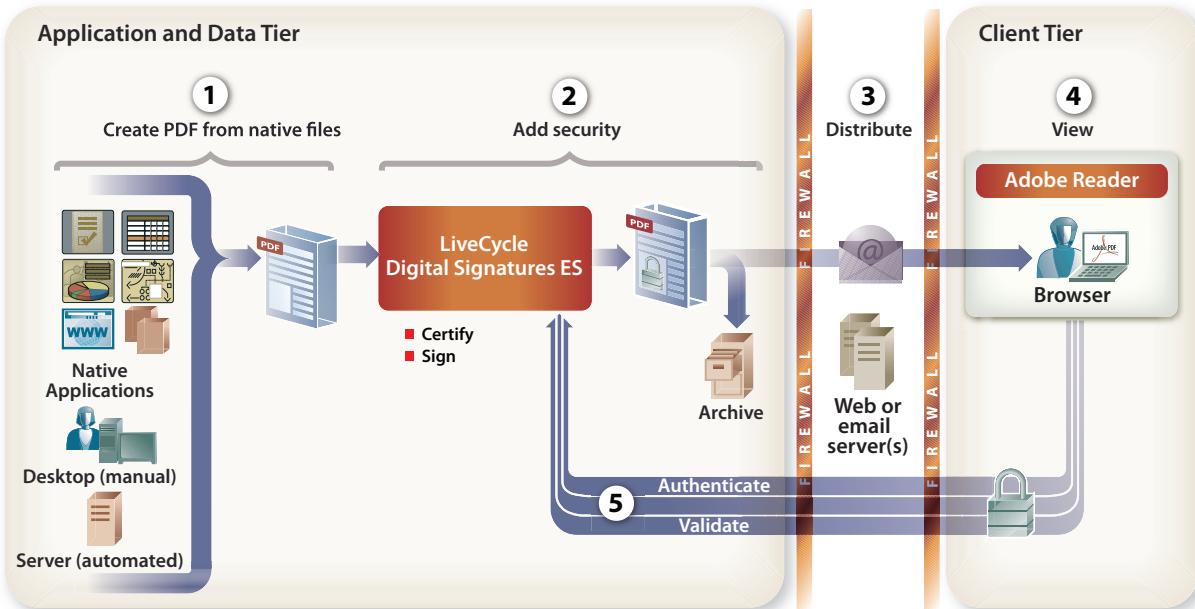
Certification and Approval signatures: Specify digital signing of documents so that recipients can validate the authenticity and integrity of the content. Digital signatures can be applied individually or in batches by using digital certificates from third-party vendors. With digital signatures applied, documents maintain authenticity even when archived.

Signature validation: Specify signature validation so that your organization can verify the authenticity of documents it receives. When digitally signed documents are received, Digital Signatures ES can open the document and validate it based on its signature status.

How Digital Signatures ES secures a document

In a typical Digital Signatures ES process, a developer creates an application that retrieves a PDF document from a specified repository, applies a digital signature by using a credential (private key) in a specified keystore, encrypts the document with a password, and sends the document to several specified recipients by email. In another example, a custom application created by using the Java API may get a series of documents, apply a digital signature to all of them, and distribute them online through the web to a number of specified locations.

The following example shows a typical Digital Signatures ES process.



1. PDF files are created using any type of PDF production method; for example, automatically from a server using Forms ES, manually on a desktop using Acrobat, or using the PDF creation capabilities of third-party applications. A client application retrieves a PDF document from a specified repository. The PDF is passed to Digital Signatures ES.
2. Digital Signatures ES secures the PDF document by certifying or signing it. It can archive the document in its secure state, as well as pass it to a web or email server for distribution.
3. A web server or email server distributes the secure document by posting it to a web site or sending it by email to recipients.
4. Only the intended recipient can open and view the secure PDF document. The recipient uses a public key (digital certificate) to decrypt the document or validate the signature. If required, the recipient can add information to the PDF document, or fill it out if it is a form, and then re-sign it for submission to the sender.
5. Digital Signatures ES receives the submitted PDF document, decrypts the contents, and validates the signature to ensure the integrity and authenticity of the document.

Services included with Digital Signatures ES

Digital Signatures ES includes these services:

- Signature
- Encryption
- PDF Utilities
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

Adobe LiveCycle Rights Management ES ensures that the business-critical information you distribute is exposed only to the people you intended. You control how people can use the information to prevent it from circulating beyond your reach. Your control over the use of information continues even after you distribute the information.

Key features

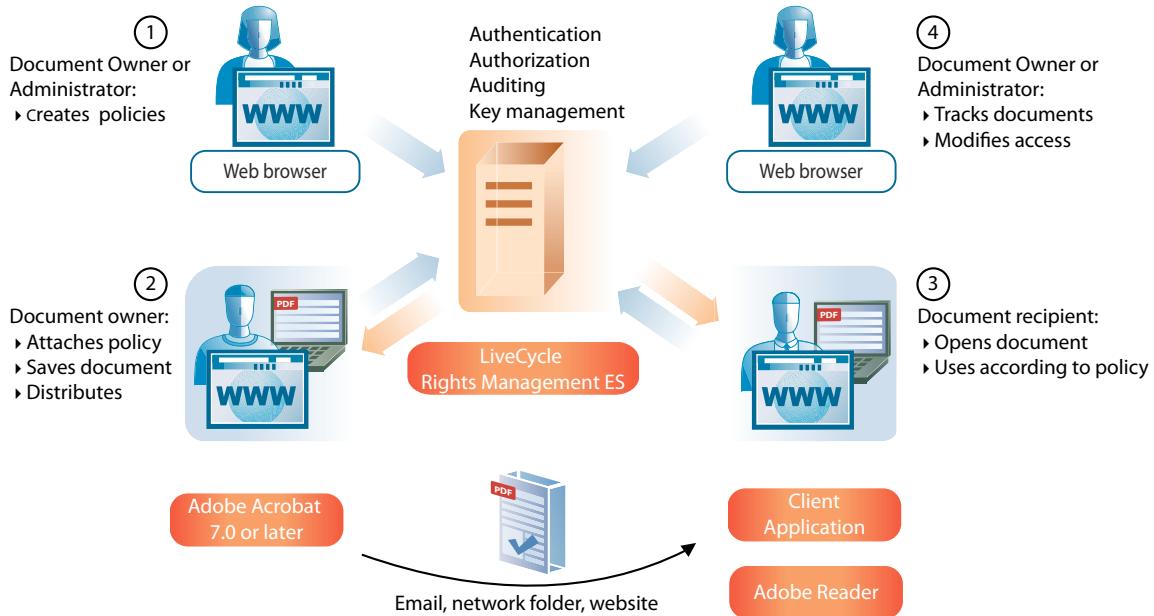
Using Rights Management ES, you can protect PDF, Microsoft Word, Microsoft Excel, and CATIA documents by using confidentiality policies. A *policy* is a collection of information that includes document confidentiality settings and a list of authorized users. The confidentiality settings you specify in a policy determine how a recipient can use documents to which you apply the policy. Because PDF documents can contain any type of information, such as text, audio, and video files, you can use Rights Management ES to safely distribute any information that is saved in a PDF document.

You can use policies to do these tasks:

- Specify who can open policy-protected documents. Recipients can belong to your organization or can be external to your organization. You can also specify different confidentiality options on the same policy for different users.
- Specify the document confidentiality settings. You can restrict access to various Acrobat and Adobe Reader features, including the ability to print and copy text, make changes, and add signatures and comments to a document. Administrators can also specify some additional confidentiality options, including the ability of a recipient to view a document offline and the ability of the user who applies the policy to revoke the document access rights or switch the policy.
- After distributing a policy-protected document, you can monitor and revoke access to the document, switch the policy, and change the access and confidentiality settings. Users can change confidentiality settings in policies they create. Administrators can change any organizational or user-created policy.

How Rights Management ES secures a document

The following illustration shows different users interacting with policy-protected PDF documents and Rights Management ES. (If Rights Management ES protected other file types, such as a DOC file, users could interact with these file types in the same manner.)



Here is an example of how Rights Management ES works:

1. The document owner or administrator creates policies by using the Rights Management ES web-based application accessible through LiveCycle Administration Console. Document owners can create user policies accessible only to them. Administrators can create organizational policies within policy sets that are accessible to permitted users, and can also designate policy set coordinators. The policies are stored in a database that connects to the application server.
Developers can also automate the creation of policies by using the LiveCycle Process Design perspective in Workbench ES or the Rights Management ES API.
2. The document owner applies the policy, and saves and distributes the document by using the web pages or Adobe Acrobat 7.0 or later. The document can be distributed by email, through a network folder, or on a website.
Developers can also automate the application of policies to documents and the distribution of these documents to end users by using the LiveCycle Process Design perspective in Workbench ES or the Rights Management ES API.
3. The document recipient opens the document in Acrobat 7.0 or later or Adobe Reader 7.0 or later. The recipient can use the document according to its policy.
4. The document owner, policy set coordinator, or administrator can track documents and modify access to them by using the web pages. Developers can also track documents by using the LiveCycle Process Design perspective in Workbench ES or the Rights Management ES API.

About Rights Management ES security

To ensure the confidentiality of documents that are protected by policies, Rights Management ES implements three layers of security:

- Authentication
- Authorization
- Document confidentiality

Authentication

All users are required to log in to interact with Rights Management ES. Users can log in through Acrobat or through the Rights Management ES web application.

Rights Management ES supports two methods of authentication:

- Username/Password. Users are prompted for their user name and password.
- Kerberos (from Acrobat on Microsoft Windows only). Users of Acrobat or Adobe Reader for Windows can be transparently authenticated.

After users are initially authenticated and, when Rights Management ES receives subsequent messages from clients, Rights Management ES uses Security Assertion Markup Language (SAML) authentication assertions to verify the identity of the message sender.

Authorization

Rights Management ES uses a role-based model to control access to the web application features. Roles also determine whether users can protect documents with policies through Acrobat.

Rights Management ES implements these roles:

Administrators: Have complete access to the server configuration and can manage all aspects of policies, policy-protected documents, external users, administrator accounts, and event audits.

Users: Can create and manage their own policies, the policy-protected documents that they distributed, and the events that are associated with those documents.

External users: Can create a Rights Management ES user account when an administrator explicitly invites them or when they are added to a policy.

Document confidentiality

Rights Management ES uses several technologies to protect documents and to provide access to them.

In general, Rights Management ES uses a symmetric cryptographic key system for encryption. Client applications such as Acrobat perform document encryption. Documents are never sent to Rights Management ES. Rights Management ES encrypts policies and licenses that are associated with documents.

The method used to protect documents depends on whether the policy requires users to access documents while online or whether the policy enables offline use.

Services included with Rights Management ES

Rights Management ES includes these services:

- Rights Management
- Encryption
- PDF Utilities
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

Adobe LiveCycle PDF Generator ES let you generate PDF documents from many different file formats, including native file formats such as Microsoft Word, Excel, PowerPoint, Visio, AutoCAD, and Corel WordPerfect, as well as open standards such as TIFF image files, PostScript, Open Document Format (ODF), and HTML. PDF Generator ES supports output to PostScript, legacy TIFF archival systems, or it can generate searchable PDF documents from image sources.

PDF Generator ES also lets you assemble single PDF documents from several source documents or separate PDF documents into constituent pages. PDF document assembly features include repagination, resizing, addition of headers and footers, insertion or deletion of pages, and creation of a table of contents, all according to your publishing requirements.

The included services provide a flexible mechanism for extending the range of the file formats that PDF Generator ES supports, and for adding support for converting almost any type of document to PDF.

Key features

PDF Generator ES offers the following features:

- Can convert almost any file format to PDF:
 - Provides an administrative user interface to manage sources, PDF settings, file conversion types, security settings, and the importation and exportation of configuration files.
 - APIs provide the ability to add file formats to existing conversion support.
- Workflow and job ticketing language supports assembling documents, attachments, and groups of documents into PDF packages.
- Supports creation of industry standard PDF/X-1A and PDF/A 1A and B from native file formats and PostScript.
- Can run as a background service.
- Offers improved performance when converting non-Adobe documents to PDF. Performance improvements vary depending on the document and how it is being converted. However, initial testing has shown significant improvement in performance when converting non-Adobe documents to tagged PDF.
- Sets the security level for PDF files produced.
- Provides control over Adobe PDF settings. Examples of such settings include whether fonts are embedded, image compression, and color conversion.

Note: Adobe PDF settings were previously known as Distiller® parameters or job options.

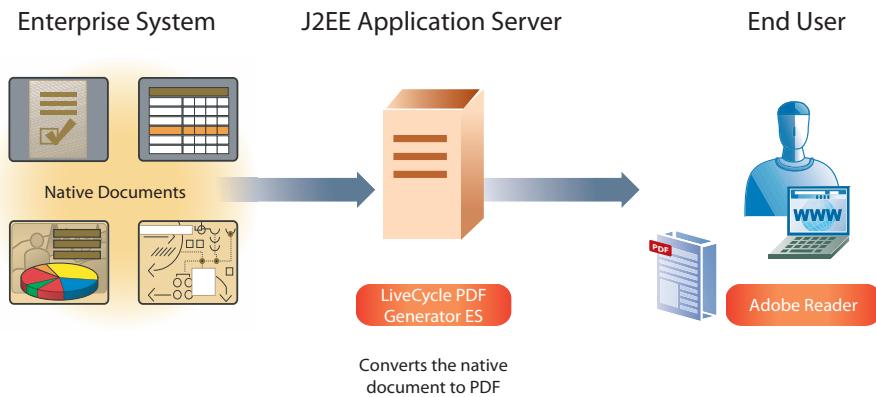
Using PDF Generator ES in the context of LiveCycle ES development tools and additional solution components, you can perform the following tasks:

- Assemble content from diverse sources and file formats into a single PDF document that includes cohesive pagination, indexing, and table of contents. You can also preserve, import, or export existing content such as annotations, file attachments, and bookmarks.
- Secure assembled PDF documents with document policies or passwords by using Rights Management ES or Digital Signatures ES

- Route assembled and secured PDF documents to stakeholders for approval and resubmission
- Embed PDF documents in automated enterprise document workflows that store information in an enterprise content management (ECM) system or archive solution. For example, you can use PDF Generator ES to convert customer letters to PDF before you store them in a document archive system.
- Provide enterprise end users access to a centralized service for PDF creation, through email or web-based interfaces, or drop (watched) folders.

How PDF Generator ES works

Here is an example of how PDF Generator ES works.



1. An administrator accesses LiveCycle Administration Console to customize the default PDF settings, security settings, and file type settings that control how PDF Generator ES converts documents. The administrator also configures PDF Generator ES to periodically scan a folder, called *watched folder*, for files to be converted.
2. A client application places a native file for conversion to PDF in the watched folder.
3. LiveCycle ES periodically scans the watched folder for files to be converted at intervals specified by the administrator when the watched folder was configured. LiveCycle ES invokes PDF Generator ES and provides the file for conversion.
4. PDF Generator ES uses the default settings that the administrator specified in LiveCycle Administration Console to convert the native file to PDF and moves the resulting PDF document to the Out folder.

Document types suitable for conversion

PDF Generator ES can convert PostScript and Encapsulated PostScript (EPS) documents to PDF, native file formats to PDF, and PDF documents to other formats.

Postscript and EPS to PDF

Using PDF Generator ES workgroups can convert PostScript files to compact, reliable, and more secure PDF files over a network. Conversion from PostScript to PDF is frequently used to convert large volumes of print documents to electronic documents such as invoices and statements or to online reports such as business intelligence or portal pages to High Fidelity PDF. Converting documents to PDF also allows enterprises to send their customers a paper version and an electronic version of a document.

Native file formats to PDF

PDF Generator ES can convert many native file formats to PDF. Such native file formats include all Microsoft Office types, including Microsoft Word, Microsoft Excel, and Microsoft Visio. Support for the Open Document Format (ODF) through the use of OpenOffice increases the scalability of PDF Generator ES. For a complete list of the native file formats that PDF Generator ES supports, see ["Input file formats" on page 52.](#)

Using PDF Generator ES, you can add support for native file formats in addition to those already supported. You can also determine how LiveCycle PDF Generator interacts with supported applications.

For more information, see *LiveCycle PDF Generator ES Administration Help*, accessible in LiveCycle Administration Console.

PDF to other format

Using PDF Generator ES, you can convert PDF into other file formats, which allows you to do these tasks:

- Convert PDF files to a print-ready format (PostScript) without manually invoking Acrobat
- Convert PDF files to TIFF images

For information about the output file formats that are supported, see ["Conversion from PDF to other formats" on page 53.](#)

Input file formats

The following file formats are supported:

- PostScript and Encapsulated PostScript (EPS)
- Image formats (JPG, JPG 2000, GIF, BMP, TIFF, PSD)
- Microsoft Word, Microsoft Excel, and Microsoft PowerPoint (DOC, XLS, PPT)
- Microsoft Project (MPP)
- Corel WordPerfect
- Plain text and rich text (TXT, RTF)
- Web (HTML)
- AutoCad (DWG)
- Microsoft Visio (VSD)
- Open Document Format (ODF) from OpenOffice
- PDF (For information about the output that LiveCycle PDF Generator can produce, see ["Conversion from PDF to other formats" on page 53.](#))

Open standards support

The following output file formats are supported when the input file is other than PDF:

PDF

PDF/X-1a: When a document is converted using the Adobe PDF setting called PDFX1a 2001, PDF Generator ES produces PDF documents that must be checked by or that must conform to PDF/X-1a:2001, an ISO standard for graphic content exchange. PDF documents can be opened using Acrobat or Acrobat Reader 4.0 and later. PDFX 1a is specified in ISO 15930-1.

PDF/X-3: When a document is converted using the Adobe PDF setting called PDFX3 2002, PDF Generator ES produces PDF documents that must be checked by or that must conform to PDF/X-3:2002, an ISO standard for graphic content exchange. PDF documents can be opened using Acrobat or Acrobat Reader 4.0 and later. PDFX3 is specified in ISO 15930-1.

PDF/A-1B:2005: When a document is converted using the Adobe PDF setting called PDFA-1B, PDF Generator ES produces PDF documents that conform to the ISO archival standard for the long-term preservation of electronic documents (called PDF/A-1B). PDF/A-1B is specified in ISO 19005-1.

Searchable PDF: This feature is especially useful in converting image file formats such as TIFF. The conversion process uses optical character recognition (OCR) to associate character shapes with the characters they represent. When the resulting searchable PDF file is opened in Acrobat, the document has the same appearance as the original. However, now you can select the characters (using the text tool) and you can search for characters.

Conversion from PDF to other formats

These output file formats are supported when the input file is a PDF:

- Image formats (JPEG 2000, TIFF, and PNG)
- PostScript and Encapsulated PostScript (EPS)

Services included with PDF Generator ES

PDF Generator ES includes these LiveCycle ES services:

- Generate PDF
- Convert PDF
- Distiller
- Assembler
- Encryption
- Distiller
- PDF Utilities
- XMP Utilities

For more information about the services included in this solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.

The LiveCycle ES connectors for enterprise content management (ECM) provide two solution components:

- Adobe LiveCycle ES Connector for IBM® FileNet
- Adobe LiveCycle ES Connector for EMC Documentum

LiveCycle Workbench ES integrates with an out-of-box repository or can be configured with ECM systems to manage assets such as forms, fragments, images, and XML schemas. LiveCycle ES connectors for ECM offer an efficient development environment. Other LiveCycle ES solution components can produce and consume the objects and their metadata to and from the connector services, therefore connecting the ECM system to the engagement application. Developers can leverage contents in ECM systems when developing applications and leverage LiveCycle ES services within the application.

To help customers develop their applications, LiveCycle ES connectors for ECM include samples for customers to quickly understand the component and develop an engagement application. Samples demonstrate how to access, render, and submit forms through an ECM client application or “front end”, such as EMC Documentum Webtop and IBM FileNet P8 Workplace.

Key features

LiveCycle ES connectors for ECM offer the following features:

- Provide integrated content repository services to other LiveCycle ES components
- Extend Workbench ES to access content assets stored in an ECM content repository, providing a unified development experience
- Provide improved performance and scalability
- Enable flexible deployment
- Enable unified invocation methods through expanded user interface choices

Integrated content repository services

Customers can develop LiveCycle ES engagement applications that process and consume contents stored in ECM systems. LiveCycle ES connectors for ECM provide integrated content repository services, including content library services, version control, and secure access to the entire LiveCycle ES product. In addition, through the component services, the connector components provide basic content repository services, such as check-in, check-out, and content metadata access.

Workbench ES developers can easily map values from other LiveCycle ES solution components, such as Forms ES, Reader Extensions ES, or PDF Generator ES, to the attributes of an ECM object. LiveCycle ES connectors for ECM make it possible to reuse and scale the same LiveCycle ES process to handle forms in different folders of the ECM system. In addition, the connectors also include service components to support ECM run-time integration.

Extending Workbench ES

LiveCycle ES connectors for ECM provide a unified development experience by extending Workbench ES to access and manage content assets that are stored in an ECM content repository. With the connectors, developers can connect Workbench ES to their enterprise content repository, develop LiveCycle ES applications using content assets in the repository, and then manage them directly without leaving the development tools.

Collaboration is easy because developers can create, modify, and store the content assets directly in the ECM repository and interact with the contents by using the standard drag-and-drop functionality.

Improved performance and scalability

The LiveCycle ES connectors preserve existing content schemas defined in an ECM system. By protecting the investment made in the ECM infrastructure, customers have choices for designing schemas to achieve higher performance and scalability of their ECM application.

Flexible deployment

Running the connectors on the same application server and operating system as the ECM system is no longer necessary. The LiveCycle ES connectors for ECM are deployed with the other LiveCycle ES components. Flexibility exists because the operating system and application server requirements are independent of those required for the ECM server.

Unified invocation methods

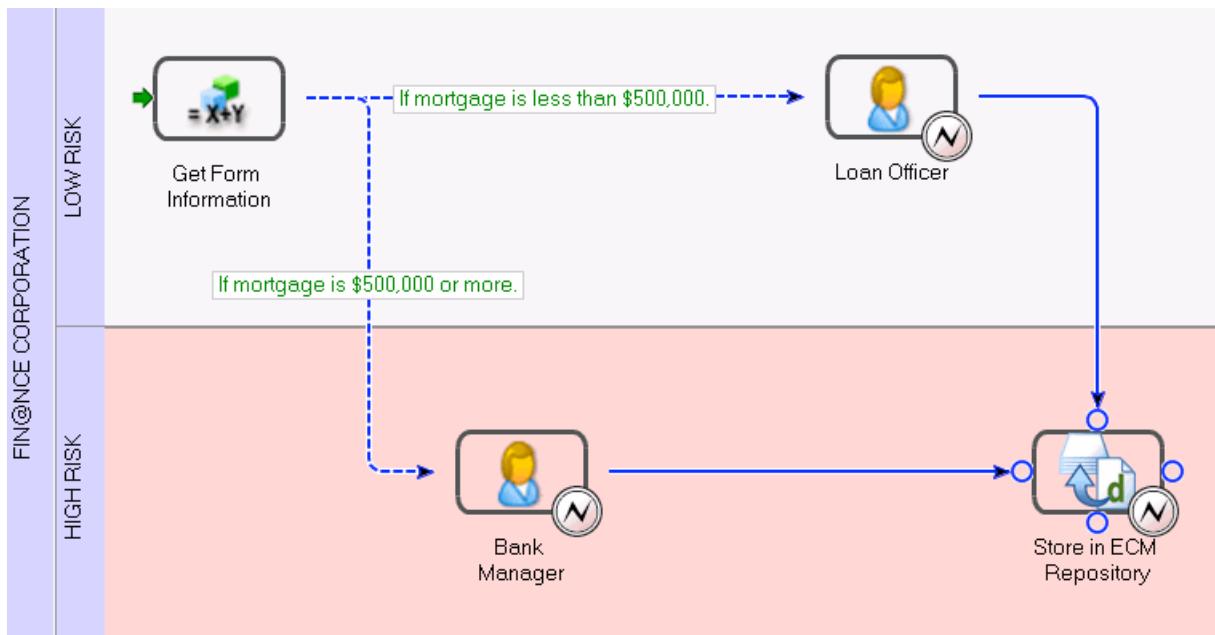
By using the expanded user interface choices, developers can develop applications using a variety of client interfaces, including ECM user interfaces such as Documentum Webtop or FileNet Workplace, portals, a Flex or desktop application, or Workspace ES.

Developers can programmatically invoke the content repository connector services and repository provider services using web services, LiveCycle Remoting, and Java APIs. They can also invoke the content repository connector services using watched folders and email invocation methods, which are available for calling these services within LiveCycle ES processes developed in Workbench ES.

How the LiveCycle ES connectors for ECM work

In a typical process that interacts with an ECM system, a developer creates a LiveCycle ES application that retrieves a document from the ECM repository and stores content in the repository.

The following example shows a mortgage application that retrieves a form from an ECM repository and stores the final document in the ECM repository.



Here is how the mortgage application works in this example:

1. A form author or form developer creates the form design in Workbench ES.
 - The form author or form developer navigates to the ECM content tree within Workbench ES.
 - The form author or form developer searches the ECM content and then uses the drag-and-drop functionality to add images and fragments to the form design.
 - The form author or form developer saves the form design in the ECM system.
2. A developer creates the business logic and process for the form. The first step in the process uses a variable that retrieves a form from the ECM repository and saves the data in a variable that a user submits from a client application. For example, a user can initiate the process from Workspace ES and submit the data by clicking a button on the form.

Based on the amount of the mortgage specified by the applicant, the form data is routed to the appropriate individual for approval. After the form is approved, as part of the last step in the process, the form data is saved in the ECM repository.

Services included with connectors for ECM

The connectors for ECM include these services:

- Content Repository Connector for EMC Documentum
- Repository Provider for EMC Documentum
- Content Repository Connector for IBM FileNet
- Repository Provider for IBM FileNet

For more information about the services included in each ECM solution component, see *LiveCycle ES Services* at http://www.adobe.com/go/learn_lc_services.